TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MARZIEH ZARE

SINGLE CELL ANALYSIS OF Z RING FORMATION IN ESCHERICHIA COLI USING MACHINE LEARNING METHODS

Master of Science thesis

Examiners: Prof. Ulla Ruotsalainen, Associate Prof. Andre S. Ribeiro, and Assistant Prof. Sari Peltonen
Examiner and topic approved by the Faculty Council of the Faculty of Computing and Electrical Engineering on 7th September 2016

ABSTRACT

MARZIEH ZARE: TUT Thesis Template
Tampere University of Technology
Master of Science Thesis, 59 pages
December 2016
Master's Degree Programme in Information Technology
Major: Signal Processing
Examiners: Prof. Ulla Ruotsalainen, Associate Prof. Andre S. Ribeiro, and Assistant Prof. Sari Peltonen

Keywords: FtsZ, *E. coli*, Fluorescence Microscopy, supervised Learning, Image Processing

In *Escherichia coli* (*E. coli*), Z-ring formation precedes the assembly of the membrane that partitions a cell into two daughter cells. Interestingly, at the beginning, as these FtsZ proteins are expressed, they first preferentially locate at the two cell poles. Afterwards, once the cell nucleoid splits in two and moves to the focal points of the cell, the FtsZ proteins start forming a ring at midcell, in between the nucleoids. Finally, the ring becomes a circle, where the septum separating the nascent daughter cells forms. Despite being the focus of intensive studies over the last decades, proper understanding of the exact role of the FtsZ protein in cell division is still lacking and, for example, to date there is yet very limited knowledge concerning the mechanisms responsible for the disassembly process of the Z-ring formation following cell division. This means that a proper study of the Z-ring formation requires observing many cells over significant periods of time and frame rate by time-lapse microscopy.

In this thesis, we have made use of the most recent methods of image processing and machine learning to identify and classify the stage of ring formation from microscopy images. We, first, segmented the cell images from microscopy images using a custom-made software. Next, we performed the sample selection technique to remove biologically uninformative samples. After that, we extracted statistical features, i.e. mean and standard deviation (std), from selected samples and then the samples were labelled by a biologist. We made use of labelled data to perform straightforwardly learning and classifying tasks. Based on the presented data, we preferably applied three supervised classification methods, namely, Decision Tree (DT), Support Vector Machine (SVM), and Regularized Multinomial Logistic Regression (RMLR). A model was generated by using pairs of feature sets and labels. Then the accuracy of the model was tested using the labelled test set, which is not the same as used data in model building.

As a result, we compared the efficiency of classifiers by evaluating their performances. We found that RMLR performs better than two other classifiers. In the following, to check if the performance of the classifiers would improve, we increased the number of labelled data. Our results demonstrate significant improvement in classification performance of all three classifiers. However, The RMLR outperforms the other two classifiers. Accordingly, in the future we will use the RMLR algorithm to perform studies where the asymmetries arising from the stochasticity of FtsZ ring formations are analyzed.

PREFACE

The work for this thesis was carried out in the M2oBSI (Methods and Models for Biological Signals and Images group) led by Professor Ulla Ruotsalainen and the Laboratory of Biosystem Dynamics, led by Associate Professor Andre Ribeiro, both of the Department of Signal Processing at Tampere University of Technology.

The work presented here was carried out between May 2016 and November 2016. The author's main contribution to the novel research results presented in this thesis was the data analysis of the microscopy measurements, as well as participation in the preparation of two scientific publications (one submitted).

Thanks to all the colleagues whom I had the pleasure to work with, particularly Elaheh Moradi.

Special thanks to all those who were co-authors in the resulting research papers, and to my supervisors, Ulla Ruotsalainen, Andre Ribeiro, and Sari Peltonen, for their patience and guidance.

Thanks as well to Leonardo Martins and Ramakanth Neeli-Venkata for their most valuable day-to-day advices regarding the methods and the software tools as well as the biology of the processes studied.

Finally, I would like to express my special thanks to my family, particularly my brother Alireza Zare and my sister Razieh Zare, for being so supportive over the years.

Tampere, 20th of December, 2016

Marzieh Zare

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

E. coli	*Escherichia coli*
μm	Micro meter
DNA	Deoxyribonucleic acid
DAPI	4’,6-diamidino-2-phenylindole
IPTG	isopropyl-β-D-1-thiogalactopyranoside
LB	Luria-Bertani
GFP	Green fluorescent protein
MAMLE	Multi-resolution analysis with maximum likelihood estimate
CCD	charge-coupled device
ROS	reactive oxygen species
ML	Machine learning
PCA	Principal Component Analysis
DT	Decision Tree
CART	Classification and Regression Trees
LR	Logistic Regression
RMLR	Regularized Multinomial Logistic Regression
SVM	Support Vector Machine
CV	Cross Validation
ACC	Accuracy
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SEN	Sensitivity
SPE	Specificity
ROC	Receiver Operating Characteristic
AUC	Area Under ROC
FPR	False Positive Rate
CNN	Convolutional Neural Network

1. INTRODUCTION

Cell division is an essential process for ensuring the continuation of survival of all living organisms. In this event a mother cell is divided into two daughter cells by following an appropriate trajectory to make sure that the offsprings are similar to the mother cell. Cell division in *Escherichia coli* (*E. coli*) is organized by a large protein complex called the divisome, which is a dynamic hyperstructure [1]. Among the encoded proteins, FtsZ is the one that acts from the start of septation by forming FtsZ ring. This FtsZ protein is required until the final step of division.

Presently, it is believed that the accuracy in symmetry in the process of cell division in *E. coli* results from the existence of two processes that are combined to obtain the desired effect. One is nucleoid exclusion [2], [3] while the other are the well-known MinCDE oscillations [4], [5]. While nucleoid-exclusion prevents Z-ring formation in the regions occupied by these dense structures, the Min system inhibits formation of Z-rings at either of the cell poles [6].

When the FtsZ gene is fused with a green fluorescent protein (FtsZ-GFP) to visualize its spatial dynamics [7], [8], one observes three apparent stages over time [9], [10]. At the beginning, once FtsZ proteins are expressed, they preferentially locate at the poles. Then, they form two dots at the cell center, located at opposite sites along the minor axis (open ring state). Finally, a circle ring is generated at the cell center, where the septum separating the daughter cells forms (closed ring state). In case the temporal-spatial organization of FtsZ is noisy, which means existence of timing and location differences between cells, its study requires observing many cells by time-lapse microscopy. One important contribution to a better understanding of Z-ring formation is the study of this process using image processing and machine learning techniques. Namely, the goal is to gain information unbiasedly from many cells and then assess at which stage the Z-ring formation is, since depending on the stage, its behavior will differ significantly when subject, e.g., to perturbations.

With that aim, here, from data that consists of confocal microscopy of FtsZ-GFP expressing cells, we use tailored image processing techniques and test various machine learning techniques for automatically segmenting and then classifying cells from microscopy images according to the stage of formation of the FtsZ ring in the cell. Since the dataset used in this work is obtained from time-lapse microscopy, the first goal of this work is sample selection, which leads to improve performance of the classifiers. The second goal is feature extraction, which is performed to gain biologically informative and non-redundant information from selected samples, and also to reduce the dimension of images. The third goal is to label selected samples by a biologist, according to the description of the stages of the FtsZ ring formation. The fourth goal is to apply multiclass classifiers to classify samples using three different supervised methods.

The following chapter presents the background to *E. coli*, cell division, stages of the FtsZ ring formation, and machine learning field. In Chapter 3, we introduce the materials and methods used in this work. This chapter also consists of theoretical presentation of used

supervised methods followed by a description in different performance measures in classification tasks. In Chapter 4, we present the results of this work. Chapter 5, contains the discussion, conclusion.

2. BACKGROUND

2.1 Escherichia coli

Escherichia coli, which is the model organism used in this thesis, is a non-spore, rod shaped, facultative anaerobe, and is a Gram-negative bacterium. In normal environmental conditions, this bacterium is 2.0 μm long and 0.25-1.0 μm wide [11].

There are several important growth factors (e.g. time of culture harvest, composition of the cultivation media, temperature, pH, etc.) that need to be optimized [12], [13], in order for this organism colonies to achieve optimal growth.

Although *E. coli* cell's preferred environment is the lower intestine of warm-blooded animals and, thus, it grows optimally at 37°C, it will not die if located outside such bodies or if the temperature of the environment is radically changed. However, the growth rate of the cells will be severely slowed down. In a laboratory setting, *E. coli* cells can be fed easily and cheaply and they grow relatively fast [14], [15]. Also, they are quite robust to genetic manipulations. For these reasons it has become a model organism for studies of gene expression.

Consequently, there have been a wide range of studies where genes are introduced by the use of plasmids. This has the intention of, e.g. achieving higher-than-natural levels of protein expression [16], [17]. One significant application of *E. coli* cells was its use to produce human insulin in large amounts [18]. For this, the cells were genetically manipulated using recombinant DNA technology. Similarly, these cells were the ones used in ground breaking studies that led to a better understanding of bacteriophage genetics [19]–[21] and thereby of viruses' genetics as a whole.

More recently, and supported by the rapid advancements in cell microscopy, genetically modified *E. coli* cells, such as the DH5α-PRO strain [22], [23], have been used the supporting organism for studies of the physical properties of cells' cytoplasm [22], [24]. They were also used in studies regarding where cellular components, such as plasmids and ribosomes, locate during a cell cycle [25], [26].

Similarly, these organisms have been used to investigate the processes leading to the aggregation of protein molecules in live cells [27], [28], the segregation of unwanted aggregates to the cell poles [29], [30], and cell division [5], [31], [32], among others.

Following this strategy, and taking into consideration the existing constructs of fluorescent proteins for this organism, we use *E. coli* as the model organism in our research.

2.2 Cell division

Cell division, or cytokinesis, is an essential process that takes place in the cell cycle of prokaryotes. In this event, a mother cell divides into two daughter cells, following a tailored process that has evolved to ensure that the progenies are as similar as possible to the mother cell, at least in what concerns the genetic material. For this to be possible, division is tightly regulated in both time as well as in space. Namely, the cell needs to ensure that the division only occurs at the proper time (e.g. only after the chromosome has been replicated) and at the predicted position, so that both mother and daughter cells retain the desired materials [33].

E. coli cells, because of being rod-shaped, grow by elongating along their major axis when under stable growth conditions. Meanwhile, they have little to no variation in width from one generation to the next [34], [35]. At a certain stage of elongation, the assembly of the constriction plane that defines the future point of cell division [34], [36] is initiated by a septum, almost precisely at the midpoint of the major cell axis [37], [38].

The event of cell division, that results in two morphologically identical cells, each with a copy of the chromosome of the mother cell, is programmed in such a way that it occurs at a specific cell length [39], [40]. Because of this, the event of cell division in these organisms is considered to be a largely deterministic process, since there is a very little variance where the point of division is located as well as when the division takes place [34], [41].

Cell division in *E. coli* is arranged by the action of at least ten proteins [42], which clearly demonstrates that this is both a highly regulated as well as a significantly complex process. Recent studies have made substantial progress in better understanding how these proteins assemble at the cell septum [8], [43]. Currently, it is believed that the expression of these proteins is controlled by temperature-sensitive genes (fts), namely, ftsA, ftsI, ftsK, ftsL, ftsN, ftsQ, ftsW, ftsZ, and zipA, etc. [8], [44]. FtsZ is believed to act from the start of septation by forming the FtsZ ring [8], until the end of the entire division process where it disassembles. Because the FtsZ protein is required from start until the final step of cell division, it is one of the best characterized and most thoroughly-studied cell division proteins [8], [42], [45].

Another important set of proteins that is part of the arsenal of *E. coli* to achieve cell division is the MinC, MinD and MinE set of proteins, also known as the ‘Min system’ (Figure 1) [45], [46]. These three proteins create a dynamic oscillation along the major cell axis that inhibits FtsZ ring formations prior to division and at the wrong places (namely, at the cell poles). In particular, the Min system produces a dynamic distribution of Min proteins whose minimum, at midcell, is used as a signal by the FtsZ proteins to assemble there and form the division site in between the two formed nucleoids [46].

More recently, single cell level observations have led to suggestions that the degree of variability of the location of the minimum in MinD concentration is too high in order to explain the very high accuracy of the degree of symmetry in cell division [46]. Something else should thus be contributing to this level of symmetry.

Based on this and subsequent studies, it was suggested that, while the Min system is indeed the one responsible for placing the division point far from the cell poles [45], it is the process of volume exclusion due to the presence of the two nucleoids at the focal points along the major cell axis that confers the additional degree of precision, i.e. symmetry, that is observed in the division process of *E. coli* cells [3]. Subsequent studies making use of live single cell imaging confirmed these predictions, as described below.

This co-operation between the Min system and the volume exclusion mechanism, that is caused by the high-density nucleoids at midcell, is nowadays believed to be what ensures the accurate symmetric positioning of the FtsZ-ring. This is illustrated in Figure 1, which is reprinted from [47] with permission from the Cold Spring Harbor Perspectives in Biology.

These conclusions have been supported by several additional observations. For example, it has been shown that when irregular nucleoid movements occur, they affect the angle and the position of the constriction plane and that, consequently, the division site localization is affected, leading to asymmetric division events [31], [48]. Furthermore, when visualized by microscopy at the single cell level, it is easy to note a visible co-localization between the nucleoid-free region at midcell and the division point, observed in both normal and aberrant-shaped cells [49].

Finally, recent study by [2] showed that in cells where the two nucleoids are more apart from each other, their degree of asymmetry in division is higher than in other cells. It also showed that, if the two nucleoids were asymmetrically positioned relative to midcell, the division point would be correspondingly misplaced, thus demonstrating the contribution from the nucleoids positioning to the symmetry of the process of cell division.

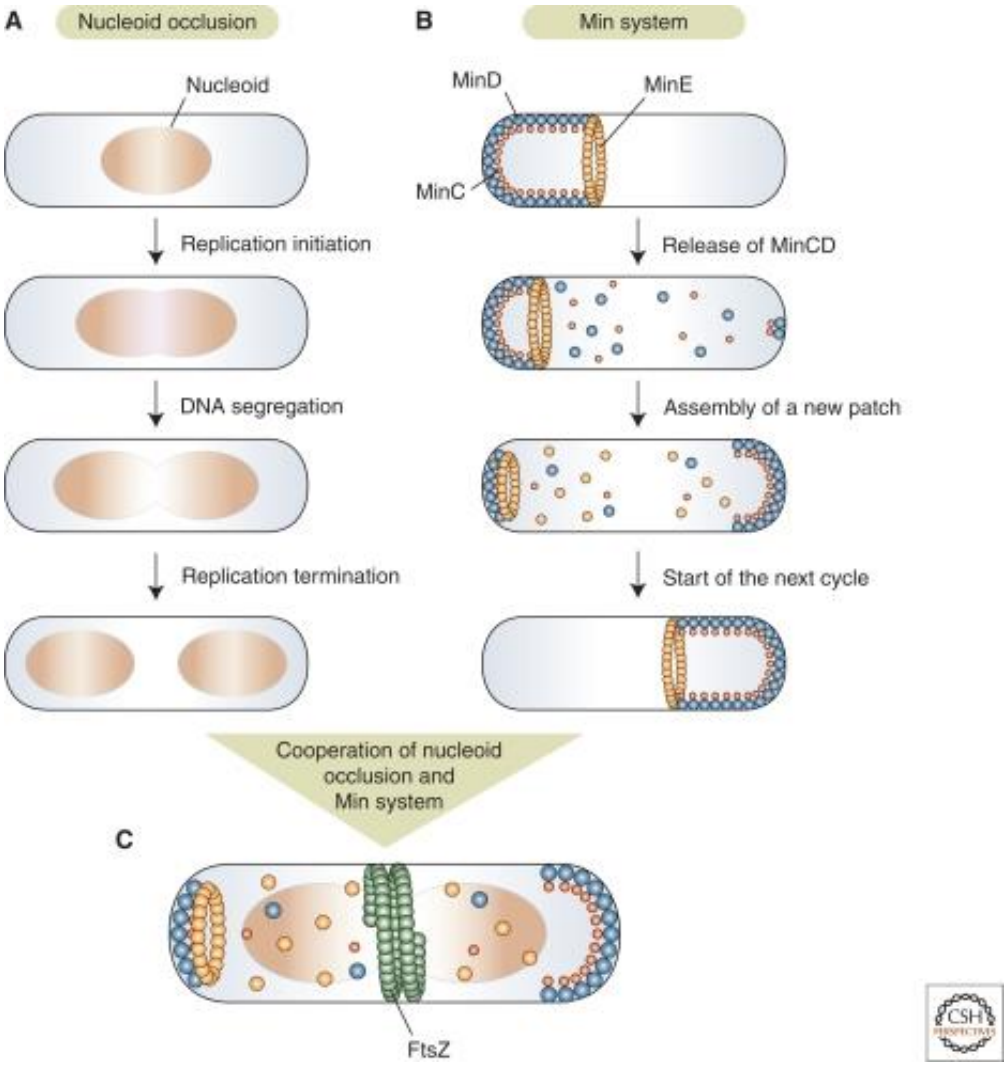


Figure 1. *The positioning of the FtsZ ring by two independent systems, namely nucleoid occlusion and MinCDE oscillation in E. coli. (A) nucleoid occlusion regulates temporal and spatial of cell division (B) The Min system inhibits the polar cell division events (C) Cooperation of the nucleoid occlusion and Min systems* [47].

2.3 Stages of the FtsZ ring formation

In general, when an *E. coli* cell divides, the placement of the division site, while based on events that are stochastic in nature, occurs so that it locates accurately at the cell center, where the Z-ring is constricted [6]. Consequently, this process generates two nearly equal-sized daughter cells, which initially have half the length of the mother. The stochasticity of the process is only visible in the fact that, in a few cases, the division site is misplaced and that, when such biasing occurs, it is irrespective of the pole age [2].

As discussed in the previous section, the accurate symmetry in cell division in *E. coli* is believed to result from the combined efforts of two independent mechanisms, namely, nucleoid exclusion [2], [3] and the MinCDE oscillations [4], [5]. While the higher density

of the nucleoids, relative to the cytoplasm, prevents Z-ring formation in the regions occupied by these nucleoids, the Min system inhibits formations at the cell poles by minimizing its signal at midcell [6].

During the cell cycle, the Z-rings exist in significantly different polymerization states [8], and these changes occur particularly during constriction. Aside from this normal behavior, under certain stresses they have been observed to be able to form and disassemble far more rapidly, ranging from 1 to 3 min for assembly and 1 min for complete disassembly [50]. This indicates that the spatial organization of these proteins can be seen as a rapid process, particularly when compared to other cellular processes, including the movement of the nucleoids to the focal points of the cell. This may in addition mean that the formations might not be very stable in nature.

When the FtsZ gene is fused with a green fluorescent protein (FtsZ-GFP) to visualize its spatial dynamics [7], [8], one observes three apparent stages during the cell cycle [9], [10]. First, (i) the cells do not exhibit any visible ring, with most FtsZ proteins being dispersed in the cell poles; (ii) next, the cells exhibit two fluorescent dots located at midcell along the major cells axis, and at opposite positions from one another along the minor axis (open ring state) and (iii) finally, cells exhibit a distinct narrow band at the cell center (closed ring state), which is indicative of the completion of the constriction wall.

Examples of cells whose FtsZ proteins, spatial distributions are in these three stages are shown in Figure 2.

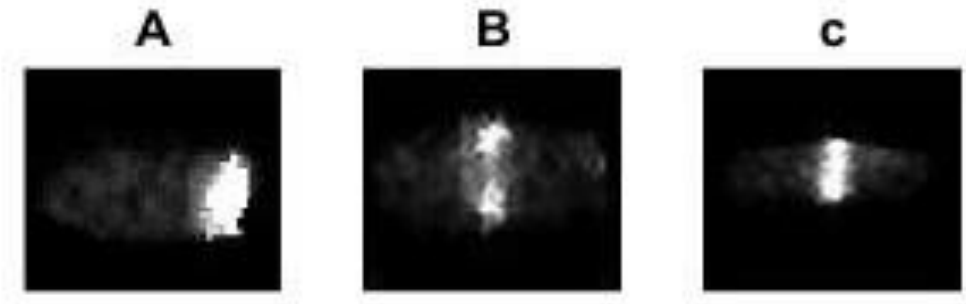


Figure 2. *Example presentation of confocal microscopy images of cells expressing Ftsz-GFP proteins at different stages. (A) cell with most FtsZ proteins at the cell poles; (B) cell in an "open ring" state; and (C) cell in a "closed ring" state.*

If one wants to study the various underlying processes of the cell wall formation based on the Z-ring dynamics using automated methods of analysis of the microscopy images, it is necessary to first create methods capable of classifying at which stage of Z-ring formation a cell is, at the moment the image was taken. Else, the problem will be too complex for present automated image analysis techniques and the outcome of the analysis will not be informative.

With the goal of finding the best automated methods of classification of the stages of Z-ring formation in individual cells from multi-modal microscopy images, we tested a variety of image processing and machine learning techniques and compared their efficiency to achieve the desired aim.

2.4 Machine learning and classification

Machine learning (ML) emerged from a question of how to engineer computer programs that can automatically improve with experience. Arthur Samuel described ML, in 1959 [51], as a "field of study that gives computers the ability to learn without being explicitly programmed". In other words, ML is programming computers to maximize efficiency and performance criteria by utilizing example data or past experience. There is a model containing a set of parameters, and learning is a process to perform a computer program to optimize the parameters of the model using the training data or past experience. The model can be defined based on the ML applications. It can be predictive that focuses on predicting in the future, or descriptive that refers to obtain more important information and knowledge from data, or both. To build mathematical models, the theory of statistics is introduced to the ML, because the major task is to gain logical conclusions from a sample.

In ML, while training the model, to assess the optimization problem and to store and process the huge volumes of data we generally have, it is required to define efficient algorithms. The representation and algorithmic solution of the learned model needs to be efficient as well. In certain applications, the efficiency of the learning becomes as important as its predictive accuracy.

Training data used in machine learning (ML) algorithms, to build a model, can be labeled or unlabeled. In labeled training data, we can build a model based on the measured variables, or response variables, while in unlabeled training data the value of the response variable is unknown. Regarding to the availability of labels in training data, there are three forms of learning, namely, supervised, unsupervised, and semi-supervised.

In supervised learning, also called 'learning with a teacher', labels of the training data are provided by a supervisor and the goal is to learn a model, using the labeled data, to gain a mapping function between training data, called the inputs, and their labels or outputs [52]. Two central problems in supervised learning are classification and regression. In supervised algorithm, there is an input, X, and an output or response variable, Y. We define a model as:

$$Y = g(X|w),$$

where $g(.)$ is the model and *w* are its parameters. In the case of regression, Y is a number and $g(.)$ is the regression function. In classification, Y is a class label (i.e. 1/0) and $g(.)$

is the discriminant function that separates the samples of different classes. The role of machine learning algorithm is to optimize the parameters of the model, *w*, such that the estimation error would be minimal, that is, minimizing the difference between estimation of the values of response variables, Y, and the correct values given in the training set.

In unsupervised learning or clustering, there is only unlabeled training data as inputs and we do not have any teacher, or supervisor. The aim of unsupervised learning is to find interesting patterns and regularities, namely similarities, in the input space. This means, the input instances are clustered or grouped based on their similarities. Due to lack of supervision for unsupervised techniques, their performance estimation is based on the subjective assessment, there are no standard methods to quantify the performance. In this case, heuristic techniques can be applied to evaluate the performance on a case-by-case basis.

In many ML applications, it is often difficult, time consuming, and expensive to provide labeled instances, as they are collected from efforts of experienced human annotators. Semi-supervised learning can be presented to assess this problem by taking both labeled and unlabeled data together into account to train a classifier. In the learning process, semi-supervised learning falls between supervised and unsupervised learning and contains benefits of both methods.

In present days, one of the most widely used techniques that has emerged from machine learning is classification. The primary goal of this technique is to predict a category or class '*y*' from a certain number of inputs '*x*'. This technique is used in a broad array of applications, including image or object classification, medical diagnosis, risk assessment, spam detection, and effective analysis.

In classification problems, in the broadest sense, the learning processes can be driven by any method that can integrate information from training patterns or from instances into the model of a classifier. The learning process is performed by algorithms that are constructed to reduce the classification error based on a set of training data.

2.4.1 Feature selection

Systems usually have a large number and wide variety of features, or variables. Also, in general, not all these features affect the process of the system that we aim to study. Because of that, in general, to study a complex process performed by a given system, we need to apply some feature selection. For this, feature selection techniques have been developed. These are techniques capable of selecting which features of the system should be used to characterize the process.

Based on their relevance, in the field of feature selection, the features of a system are usually categorized into 4 types, namely, irrelevant, redundant, weakly relevant (may be

necessary), and strongly relevant (cannot be removed). In general, a set of useful features includes all strongly relevant features, some of the weakly relevant features, and none of the irrelevant or redundant features.

When dealing with classification problems, in order to improve the classifier performance and thus obtain a greater understanding of the data based on the performance of the classification tasks, one usually starts by using some feature selection method as a dimension reduction technique [53].

3. MATERIALS AND METHODS

3.1 Chemicals

To produce the bacterial cultures, the cells were grown in Luria-Bertani (LB) medium (10 g of tryptone per liter, 5 g of yeast extract per liter, and 5 g of NaCl). In addition to this, antibiotics were added to this media, to guarantee the maintenance of the plasmid in the cell colony (purchased from Sigma-Aldrich). Next, in order to induce the expression of FtsZ-GFP, isopropyl-β-D-1-thiogalactopyranoside (IPTG) was added to the culture prior to visualization of the cells under the microscope. Finally, agarose was used for microscopic slide gel preparation.

3.2 Strain, Plasmids and Medium

The *E. coli* CM735-derived strain NK9386 was used to express the gene hupA::mCherry, under the control of the native promoter, which is incorporated into the chromosome [54] (a kind gift from Nancy Kleckner, Harvard University, U.S.A). Additionally, this strain was transformed with pEG12-*ftsZ::gfp* [7] (kind gift from Kenn Gerdes, Copenhagen University, Denmark), which is under the control of a Lac promoter, and thus requires induction by IPTG (see above). Overnight cell cultures were grown in LB media with the antibiotic Ampicillin for 15 hours at 37 °C with shaking (250 rpm), prior to being prepared for visualization under the microscope.

3.3 Induction of FtsZ-GFP Expression

To prepare cells for visualization under the microscope, overnight cultures were diluted into fresh LB media with ampicillin. These subcultures were then left in the incubator at 37 °C with shaking (250 rpm) until the cells grew to midlog phase (OD_{600} ~ 0.3). At this point, Ftsz-GFP expression was induced by adding 40μM IPTG to the culture. Cells were then left in the incubator for an additional 30 minutes prior to microscopy.

3.4 Live cell imaging methods

Fluorescence microscopy was used for the visualization of the cells, as well as of the FtsZ-GFP proteins within. The steps of the microscopy techniques, which were employed to produce the images that were analyzed and classified, is provided as follows: FtsZ-GFP proteins were chosen to be observed since, first, GFP is a highly fluorescent protein that has a self-contained fluorophore. Also, it has no known external substrate requirements other than molecular oxygen [55].

This protein is a well-established valuable tool in probing the localization of proteins within living cells of many species [56], in particular bacteria [57]. Our present results shown in subsequent sections are also supportive of this, by indicating that FtsZ-GFP is a useful tracker of the dynamics of the underlying processes of bacterial cell division.

Meanwhile, FtsZ is tracked because in our current understanding of how bacterial cell division occurs, this protein forms a ring that informs other cellular components where the division plane should locate, by creating a cytoskeletal framework for the subsequent action of other proteins such as FtsA. In support, it is well-established that FtsZ and FtsA proteins tagged with green fluorescent protein (GFP) colocalize in living bacterial cells in the space between the two segregated nucleoids prior to division.

Interestingly, under certain stresses, cells with elevated levels of FtsZ–GFP or FtsA–GFP exhibited bright fluorescent spiral tubules that spanned the length of filamentous cells. This abnormal behaviour suggests that FtsZ may form unlocalized spirals under some conditions and that FtsA can bind to FtsZ in such cases.

In the case of the data analyzed in this thesis, the cells were imaged using a Nikon Eclipse (Ti-E) inverted microscope with a 100x objective. Phase-Contrast images were acquired using a charge-coupled device (CCD) camera (DS-Fi2, Nikon). Confocal microscopy was utilized to detect IbpA-YFP aggregates (exposure using 488 nm laser) and HupA-mCherry tagged nucleoids (exposure using 543 nm HeNe laser).

Epifluorescence microscopy using a mercury lamp for UV exposure was used to detect the DAPI-stained nucleoids. Temperature controlled chamber was used to keep the microscopy slides at 37 °C during image acquisition. When capturing the time series, images from the confocal channels were acquired every minute for one hour to track the dynamics of the aggregates and the nucleoids.

To image the dynamics of Z-ring formation and its stages of development, multi-modal confocal (for detecting the ring and the nucleoids) was used as well as phase contrast microscopy (for detecting cell borders). As mentioned above, for image acquisition, FtsZ-GFP induced cells were placed on 1% agarose gel pad prepared in LB and supplemented with 40µM IPTG.

The FtsZ-GFP fluorescence was visualized under the fluorescent confocal microscope using a 488 nm argon ion laser (Melles-Griot) and a 515/30 nm detection filter. Images were acquired using a medium pinhole, gain 90 and 3.36 µs pixel dwell.

In general, cells were visualized once, 1 hour after inducing the expression of FtsZ-GFP. The confocal images and the phase contrast images were taken nearly at the same time to ensure co-localization of the cells in the two images. Phase contrast images were captured solely for the purpose of cell segmentation.

Note that even though it would be desirable to sample the FtsZ proteins locations over time to characterize their dynamics, in practice, due to the current limits of production of these proteins by our cells, this is strongly limited by a phenomenon called photo-bleaching. This phenomenon occurs because each fluorescent protein molecule has only a limited number of photons it can produce following radiation, after which it loses its fluorescence capacities [58].

Since imaging the cells multiple times would lead to higher exposure to the laser, the photon 'budget' of each FtsZ-GFP would rapidly be depleted, leading to the bleaching of these proteins, which would make the assessment of the spatial localization of the Z-rings less accurate. For this reason, the images of cells that were imaged only once were not analyzed. In any case, since the observed cells' division process is far from synchronized, by observing many cells, it was being able to collect images of cells at many stages of division, which provided a general picture of the dynamics of the process of division.

Next, we describe the microscopy methods employed in more general terms and detail [8].

3.5 Microscopy

Microscopy is a technique that uses a device for enlarging small objects so that they become visible. To do this, the device makes use of optical magnification techniques whose goal is to increase the distance between the rays of light on the plane of projection. This technique is currently a rapidly evolving technology that has become the primary tool for analyzing dynamical processes in live cells.

Modern fluorescence microscopy techniques enable the imaging of cells as well as of organelles and other subcellular structures at high frame rates in individual live cells, and even inside living animals. In this section, we describe some of the most common microscopy techniques, with special focus on those that were used to obtained the images that were analyzed here.

3.5.1 Fluorescence microscopy

Imaging individual proteins, proteins aggregates, or organelles can be done in live cells via fluorescence microscopy. For this to be possible, the cells need to contain fluorescent proteins or chemicals.

A fluorescence microscopy measurement works as follows. First, one has to illuminate a certain specimen with light (excitation light) at a specific wavelength that will cause that specimen (the fluorescent molecules, or fluorophores), to emit light (emission light). Importantly, it does so at a different wavelength than that of the excitation light, which allows distinguishing between the two, provided the proper light filters.

One fluorescence microscope technique is called epi-fluorescence microscope, which makes use of a wide-field illumination, where the whole specimen needs to be illuminated at once by filtered light emitted from a lamp. The emitted light from the sample is then filtered and has to be collected by a camera [59].

In fluorescence microscopy, the specimen is visualized with the use of a fluorescent 'label'. This label consists of a fluorophore that needs to localize in the region to be observed, such as the cytoplasm of the nucleoid, or it needs to be tagged to the object to being observed, such as to an organelle, the DNA or a protein.

Such labels are commonly referred to as 'stains' or 'dyes'. In live-cell imaging, particularly when performing time-lapse microscopy over large time scales, it is often required that the fluorophore does not hinder cellular functions. Else, the cell behaviour might be perturbed to a level such that it no longer is a proper model of the real system. As an extreme example, if a label kills a cell after a few minutes, the observed behaviour afterwards will no longer be valuable information on how the cell behaves when alive.

The best label to use depends significantly on what information one wants to extract. A more common example is DAPI. It is a commonly used stain that binds to the DNA, and due to being easily visualized, it is therefore used for visualizing the nucleus of cells. The drawback of using this stain is that it interferes heavily with vital cellular processes, such as transcription. As such, it is useful to observe how the nucleus (or nucleoids in the case of bacteria) looks at a given moment in time but, for example, it will affect too much the experimental results of time series measurements.

Fluorescent proteins on the other hand, are ideal for time series measurements, since, in general, they interfere only weakly with the growth of the host cells [60]. These proteins were originally extracted from animals such as jellyfish, which express them naturally. It is required, however, stronger laser power, which needs to be used with care, since it can affect cells.

One commonly observed organelle is the mitochondria. In most microscopy-based studies of mitochondria, the label used is a fusion of a protein that localizes in the mitochondrial matrix or intermembrane space, and of a fluorescent protein. One common label is mitoDsRED2, which is a fusion between the red fluorescent protein DsRed2 and the mitochondrial-targeting component of the human cytochrome c oxidase. Such studies can be conducted even in live animals. For this, the animals have to be genetically modified in order to express the fluorescently labelled proteins. As an example, [61] made use of a transgenic mouse line that selectively expresses mitochondria-targeted fluorescent proteins in neurons.

The main problems in live cell microscopy are usually those associated to cell 'photodamage'. Namely, exposure to light is known to cause significant perturbations in cell homeostasis; which is referred to as photodamage. It is believed that photodamage is

mainly a result of light interacting with the fluorescent molecules, that causes them to become chemically reactive. This in turn causes the generation of reactive oxygen species (ROS) in cells [62].

In mammalian cells, well-known effects of photodamage are delayed mitosis, or a complete arrest of the cell cycle. Interestingly, the effects vary widely between cell types, as well as between individual cells. Cells can also become more sensitive to photodamage if affected by other stress factors for unknown reasons, and thus particular care is needed when imaging cells already in stress conditions [63].

When imaging spots or organelles inside cells, epi-fluorescence microscopy resolution is insufficient. Because of this, epi-fluorescence imaging is often supplemented with deconvolution methods, which attempt to increase the resolution by mathematically inverting the blurring [64]. To avoid out-of-focus illumination existing in epi-fluorescence and to enhance resolution, several methods have been developed, with the most commonly used one being confocal microscopy. Confocal microscopy reduces the focal volume, therefore reducing the out-of-focus light with a pinhole. A drawback of this is that the sample is illuminated only one volume at a time and must be scanned, which slows down the imaging. The speed can be improved with certain setups, such as using spinning-discs that illuminate simultaneously multiple regions of the sample.

The enhanced resolution of confocal microscopy is achieved by passing the emission signal through a pinhole aperture before reaching the detector, because this allows discarding much of the light from outside of the illumination volume. This method is therefore effective at optical sectioning, i.e. in being able to image three-dimensional objects as series of thin sections at different focal planes [59].

While confocal microscopy is effective for single-cell imaging, in general it is not well suited for imaging tissues. For example, the depth of the imaging is very limited (approximately 40 µm), because the tissue causes strong scattering of the light [65]. Also, the excitation light, as it passes through the tissue, causes a large amount of photodamage. For these reasons, a more suitable technique would be multiphoton microscopy, which uses multiple light pulses to excite any given fluorophore simultaneously.

This will cause emission at a wavelength higher than that of the individual pulses, which will allow imaging much deeper inside the tissue, due to the strong decrease in the degree of scattering of the light. Additionally, the excitation pulses do not need to be of as high energy as when using confocal microscopy. The photodamage in the out-of-focus regions will thus be much reduced.

3.5.2 Phase contrast microscopy

Phase contrast techniques aim to enhance small variations in phase between neighbor points in space so as to make objects more visible, by transforming these variations into differences in amplitude instead. For example, as one moves in the space of the image from a pixel with a cell to a pixel without a cell, there will be small phase changes. If these can be transformed into differences in amplitude, our eyes will more easily locate where the cell border is.

Importantly, this can be done based on visible light, which is not harmful to cells, allowing to observe them without killing or causing much damage. In general, this technique is used to complement other microscopy techniques. For example, one can use confocal microscopy to visualize proteins inside the cells, and phase contrast to detect where the cell borders are.

3.6 Image Analysis

In this thesis, we used image analysis tools to reveal relevant information from the microscopy images, which it is then used to reveal underlying biological mechanisms of these cells. For the image analysis tasks, we made use of existing custom-made MATLAB^{TM} tools, developed in succession [29]. The analysis steps and the used methods are described in more detail in the following subsections.

3.6.1 Methods of cell segmentation

The first step in the image analysis step was cell segmentation, segmenting the cells from the background. Cells were detected and segmented from the phase contrast images using a custom-made software that integrates the software 'MAMLE' [66] and 'CellAging' [29].

MAMLE (Multi-resolution Analysis and Maximum Likelihood Estimation) performs the segmentation of individual cells from the image in two steps. First, it over-segments cells by multi-resolution edge detection that makes use of information on the morphological properties of the cell (i.e. assumes a specific shape and some expected upper and lower limits of size). Next, the over segmentation is corrected. That is the various regions are merged so as to maximize the "cell likeness" objective function for all cells. This is performed by a maximum likelihood based method.

Meanwhile, CellAging is a tool that was designed to automatically extract information on the inner fluorescence of *E. coli* cells. From either fixed or time-lapse microscopy images, it performs cell segmentation, aligns the phase contrast and fluorescence images, performs lineage construction in the case of time series, and then informs on the fluorescence within the cells at any given point in time. In CellAging tool, the segmentation is based

on the Gradient Path Labeling technique [29] and uses classifiers to merge and dismiss segments. The classifiers are constructed using a Classification and Regression Trees algorithm [29], and are pre-trained by an expert [67].

By using the integrated software, first, cells are segmented automatically, the image is split into separate regions occupied by each cell, then the result of segmentation is corrected manually. During the segmentation process, the dimensions, location, and orientation, of segmented cells are measured and extracted by applying principal component analysis (PCA) technique [26]. In the following, fluorescence images (which visualize FtsZ-GFP) were automatically aligned to phase contrast images by 'CellAging' [29]. Finally, CellAging automatically aligns the fluorescence images with the segmentation results.

The output of the segmentation process comes in the form of masks, which denote the regions occupied by each individual cell. Based on these masks, the information inside is then collected depending on our goals. Here, we were mostly interested in the spatial location of FtsZ-GFP proteins and thus, on the degree of fluorescence in each cell pixel. Figure 3 describes simply the cell segmentation step that has been done in this study. When the cells were segmented automatically, by using the integrated tool, we corrected the segmentation result manually to ensure all cells have been segmented as well as possible, which leads to gain more accurate data for the classification process.

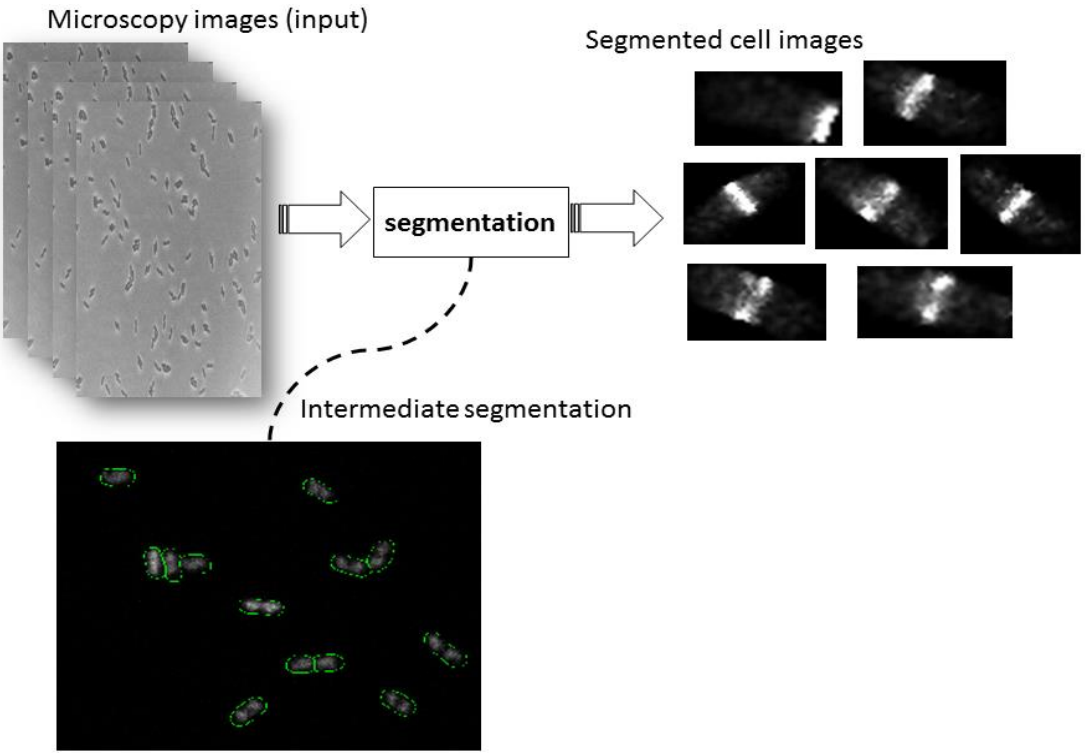


Figure 3. *Automatic cell segmentation using integrated software, MAMEL and CellAging. The final segmentation result was corrected manually.*

3.6.2 Methods of Data pre-processing

Pre-processing steps play an important role in classification problems.

First, one needs to ensure high quality data as much as possible, as it directly affects the quality of the results of classification method. The data can, e.g., be noisy or contain irrelevant information. In these particular cases, removal noise and irrelevant information can much enhance the results. These problems are usually present and are very specific in the case of biological processes. At the moment, no existing algorithm or method can cope with all issues and thus, a variety of methods have been developed [68] from which one needs to the select the best suited ones.

One mean to enhance data quality is sample selection. This usually includes outlier detection and removal steps. There are several sample selection methods. These are usually either filter or wrapper-based methods [69]. Sampling is sometimes also needed if one of the classes is underrepresented in the training data which would bias the results. Several solutions have been proposed on how to handle with this [70].

In this study, we used filtering by sample selection. This technique will always result in data reduction. By using this method, we checked each sample and removed those whose values that were above specified thresholds (e.g. representing unrealistic values). This was a complex task, since many erroneous samples can lie within the 'standard' distribution. The other reason that we performed sample selection was that it could also be used to handle with the problem of incomplete data, a common issue in classification tasks [71], [72]. Incompleteness can arise from a value being lost, etc. There are several proposed methods to cope with missing data [73]. E.g. samples with unknown features can be left out, or missing values can be replaced with the most common or mean value for that feature, etc.

If the data is from a continuous variable, the classification and learning processes are usually harder. Discretizing the samples is a common solution. Discretization algorithms can be supervised, and thus depend on class information, or unsupervised, which do not take into account the class labels [74].

Error removal and data correction are usually insufficient to achieve good classification results. For example, in general, it is better to also normalize the data if the maximum and minimum values differ in orders of magnitude or if the range of possible values differs between measurements. To do this, we normalized our data to scale to [0...1].

3.6.3 Method of feature extraction

In our case, namely, the classification of the steps of FtsZ rings formation preceding cell division, since the biosystem "cell" has an enormous number of parameters (and so do

the microscopy images of these cells) that constantly change during the division process, although, most likely, most are not directly linked to it and thus are not suited to characterize it. In this work, in order to extract features containing desirable information, form segmented cells, we, first, oriented images to fix them in the same orientation, by using the orientation values of segmented cells measured during the segmentation process. Then, we performed normalization process on the oriented images, to scale them to [0...1]. In the next step, as it is shown in Figure 4 ,we split the oriented and normalized cell images into three sections, the two poles and midcell, having these separation points located at positions 0.25 and 0.75 along the major axis (length normalized to 1).

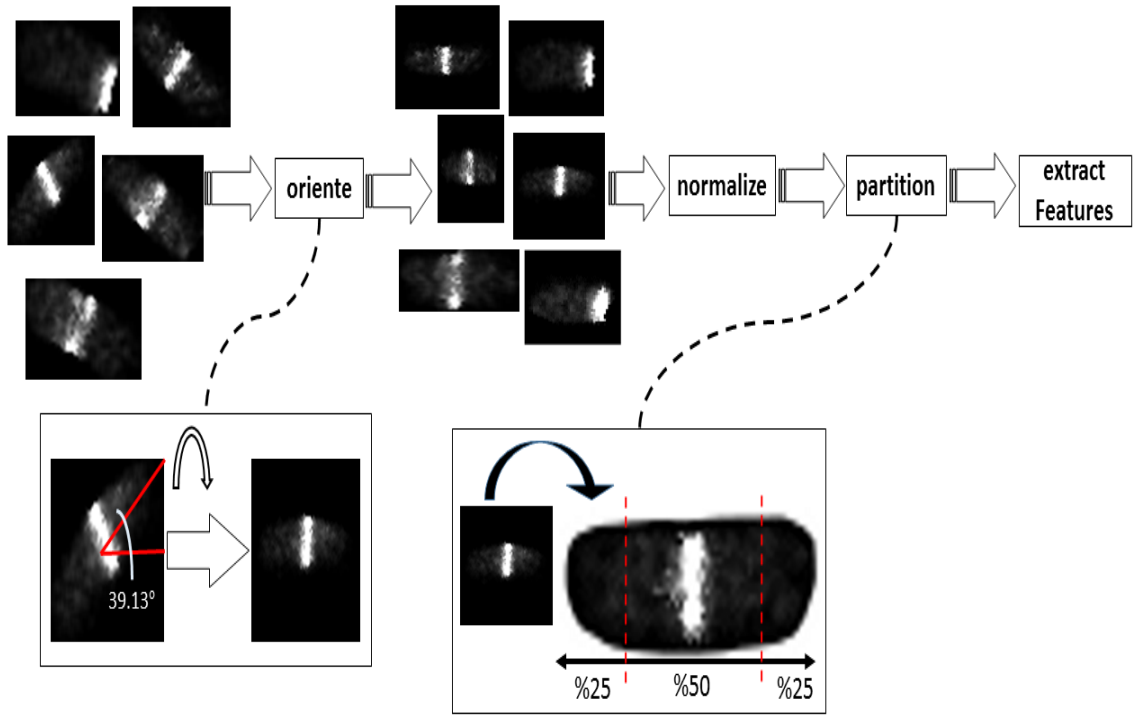
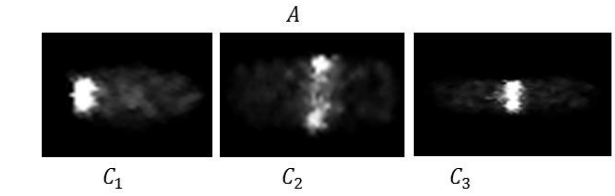


Figure 4. *The orientation of the cell images was fixed. And then, oriented images were normalized to [0,..,1] and they were partitioned into three regions.*

Finally, from the distribution of fluorescence intensity from FtsZ-GFP along the major cell axis of each cell, one example is shown in Figure 5 A, we extracted the mean, variance, standard deviation (std), skewness, and first order histogram. We observed these features and then, by pre-analysis, we selected the standard deviation and the mean as the features for classification, as they exhibited more ability to best discriminate the stages of FtsZ ring formation when compared to the other features tested, From Figure 5 C, we can see how well the selected features would be able to distinguish class from other classes.



C

	std	mean	std	mean	std	mean
C_1	0.237	0.191	0.027	0.045	0.019	0.034
C_2	0.020	0.037	0.156	0.143	0.022	0.037
C_3	0.023	0.032	0.220	0.143	0.021	0.031

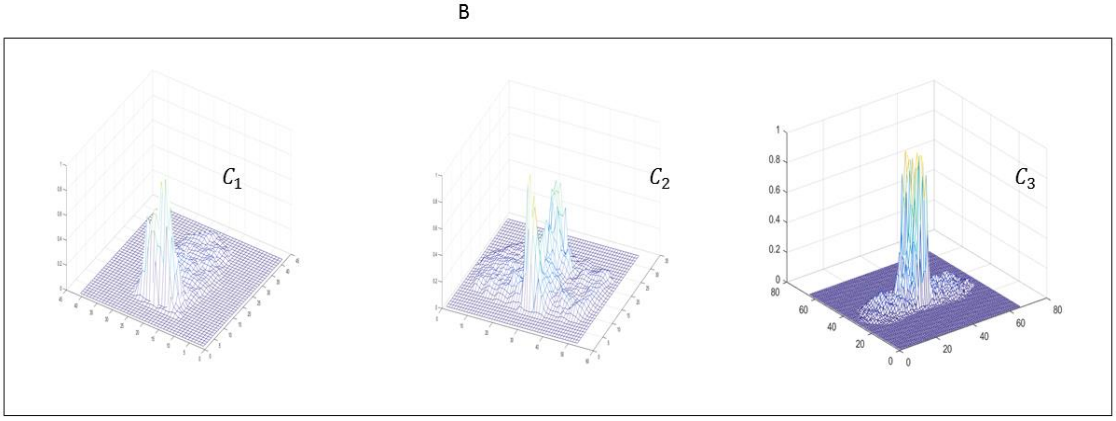


Figure 5. *Part A shows examples of each class. Figure B shows the fluorescence intensity from FtsZ-GFP along the major cell axis of the cells in each class. the Table in part C is related to an example of the measured features from cells of each class.*

3.6.4 Model selection using cross-validation

In learning problems, the models are built by using labelled data, namely training data, and the performance of the models is estimated by using a test set with known labels. In a data-rich situation, the data can be simply divided into training and test sets [75]. Since we applied supervised learning algorithms, the available data is limited. For this reason, we used cross validation as a technique to provide proper data sets, training and test, in order to build the model and estimate the performance.

In K-fold cross validation, the data is divided into *k* subsets of equal size and the model is built *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are considered as a training set. Then, the average performance across all *k* trials is calculated (Figure 6). The advantage of this method is that the variance of the resulting estimate is decreased as *k* is increased. Also, it does not matter how the data is divided. Meanwhile, its computational complexity can be regarded as one of the disadvantages, as the training algorithm has to be ran *k* times, which means it takes *k* times as much computation to make an evaluation.

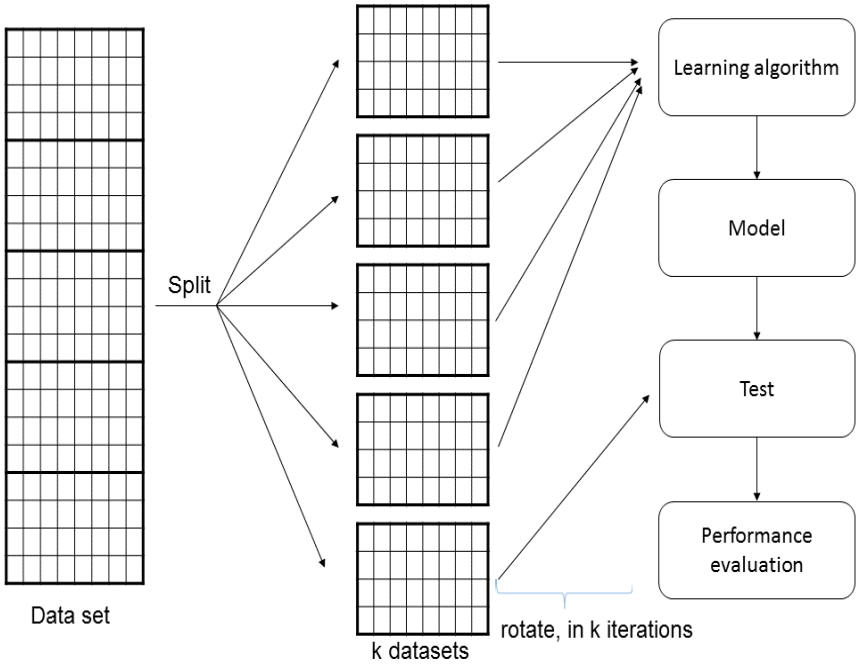


Figure 6. *Block diagram of K-fold cross validation.*

3.7 Classification methods for determining FtsZ ring formation stages

In this section, we describe the classification methods that are selected and used, as multiclass classifiers, in this study. We, first, introduce Decision Tree (DT), as a rule-based classifier. Then, we present Support Vector Machine (SVM), as a discriminative classifier. The third presented method is Regularized Multinomial Logistic Regression (RMLR), as a statistical model. All these three methods are known as supervised methods. However, they were not the only solutions for our problem, that was to classify the stage of Z-ring formation in individual cells, and the problem could be solved by using unsupervised or semi-supervised methods. The reason for using supervised methods to solve our problem was that the solution, or target values, was defined by the biologists studying this process, since they acted as a supervisor for our data. In addition, learning a model using labelled data would be easier.

3.7.1 Decision Tree

Decision tree (DT) is a statistical machine learning method and is known as a supervised learning technique. The aim of DT algorithm is to learn decision rules generated from training data, or features and predict values of response variables through those learned decision rules. Decision rules are produced as interpretable if-then-else decision rulesets which are similar to graphical flowcharts, as it is shown in Figure 7. We will explain how to build this type of tree in the following.

Decision Trees (DTs) can be utilized in both regression and classification problems. In this case, they are often called as Classification and Regression Trees (CART). DT/CART models are sometimes regarded as an example of adaptive basis function models, which are more general field of machine learning. In these models, features are not predefined, but rather are learned directly from the data. Since the parameters of these models are nonlinear, they can only be estimated by applying locally optimal maximum likelihood estimate (MLE) algorithm [76].

By applying DT/CART models, the feature space is divided into a number of rectangular regions, partition using axis parallel splits. In order to make a prediction for a given instance, one can use the mode and mean of the response variables of the training samples of the partition that the new instance belongs to.

In DT classification problems, the aim is to predict a categorical response value, or class label, using the mode of the training data of the region to which an instance belongs. That is, the most occurring class value is obtained and assigned as the response of the instance. In this study, in order to actually do a classification using DT, we considered a pre-grown tree with a Top-Down construction, as it is shown in Figure 7. The process is started at the root node of the tree. The root node asks if x_1 is less than threshold t_1. If the answer is yes, this is then asked that if x_3 is less than some other threshold t_2. If yes, we ask if x_2 is less than t_4. If yes, we are in the bottom left of space, R_1. If no, we are in the bottom right of the space, R_2. This process is then iterated for the other nodes and leafs. At the end, we would have a set of if-then-else decision rules that can be used to classify the new observation.

Figure 8 indicates an example of subset of $\mathbb{R}^2$ containing training data, when we have two feature variables (X_1, X_2). From this figure, we can see that how the domain can be divided by axis-parallel splits, each split of the area is aligned with one of the feature axes.

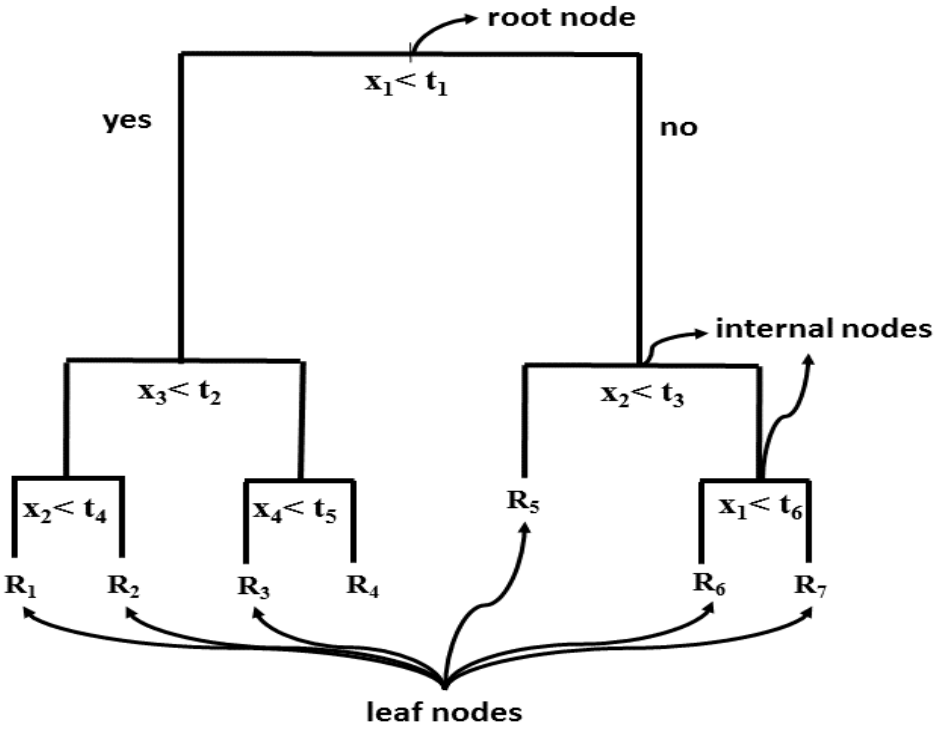


Figure 7. *A general scheme of Decision Tree.*

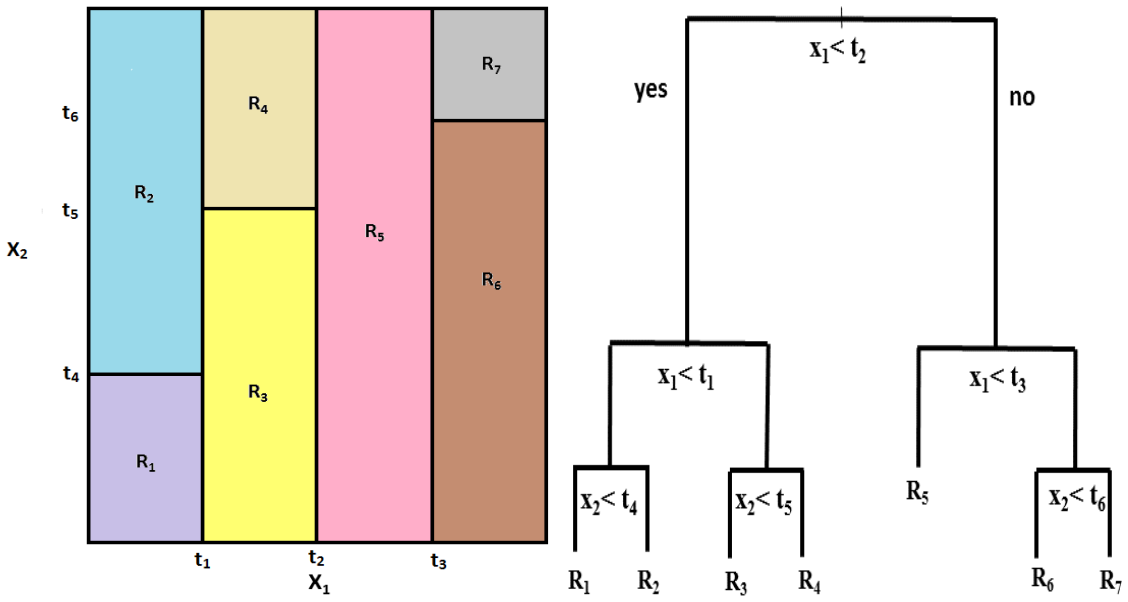


Figure 8. *Training data, in $\mathbb{R}^2$ subset, is partitioned into seven sub-spaces.*

The performance of a DT classifier always depends on how well the tree is designed, which usually is a complex task [77]. In addition, it would not be always possible to build trees that model perfectly the training data, too little data might be located within each subtree that it results overfitting. However, in order to avoid overfitting, one solution is performing pruning. This technique can be applied before growing a tree completely, which is called pre-pruning, or after growing a full tree, which is called post-pruning. In

the pre-pruning process, one can stop growing the branches of the tree that obtain unreliable information. In the post-pruning technique, first we let the tree to grow fully to gain all possible feature interactions and then replace some nodes with leafs to simplify the tree.

DT classifier can cover both binary and multiclass classification problems. In this work, we applied this method as a multiclass classifier. Figure 9 indicates the block diagram of the used DT classifier. In this case, we used cross-validation technique to partition our labeled data to training and test sets. We used only the training subset to fit the model and only the testing subset to evaluate the accuracy of the model.

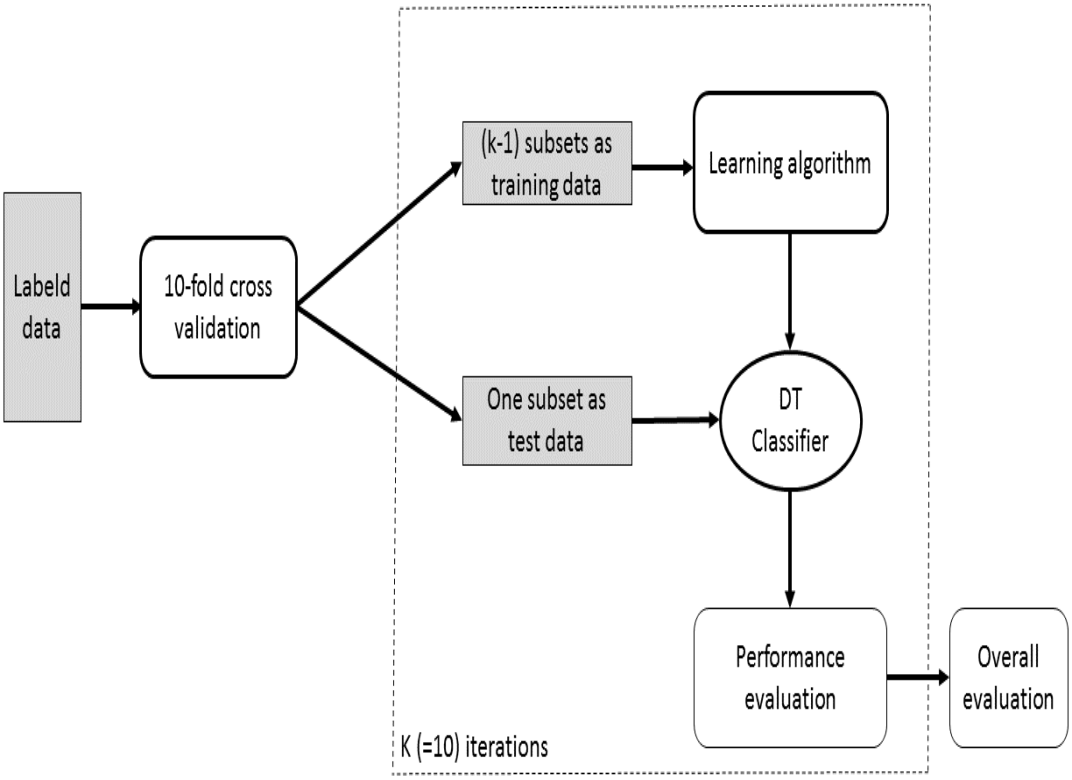


Figure 9. *Decision Tree (DT) classifier scheme.*

3.7.2 Support Vector Machines

The idea of support vector machines was invented by Vapnik in 1979 and was formulated in its current form in [78]. Support vector machines (SVMs) have recently gained a lot of popularity in machine learning applications, including character and object recognition [79], [80], face detection from images [81], and text categorization [82]. SVMs have shown significantly better or at least the same generalization performance with respect to the competing methods.

The ideas of SVMs have recently been generalized in order to connect them to other existing algorithms and theories [83]. In the literature, there are some efforts which demonstrate SVMs as a powerful technique in many biological cases as well [84], [85].

The SVM is a linear classifier that separates the classes in the training data using an optimal separating hyperplane. The general idea of SVMs is to map the data into a feature space where it becomes linearly separable [83]. This is performed by mapping the data into a higher dimension space where the classes can be separated by a hyperplane. In the linearly separable case, it can increase the margin between the classes, where the hyperplane is maximally far from the closest training sample of each class. For this reason, SVMs are often called maximum margin classifiers. The concept of margin and the optimal separating hyperplane are illustrated in Figure 10.

One of the main efforts in classifier design is the generalization of the solution. This goes along with the SVM principle of maximal margin, which states that how far larger margin is selected, the error of the linear classifier (which is determined by the separating hyperplane), can be generalized better.

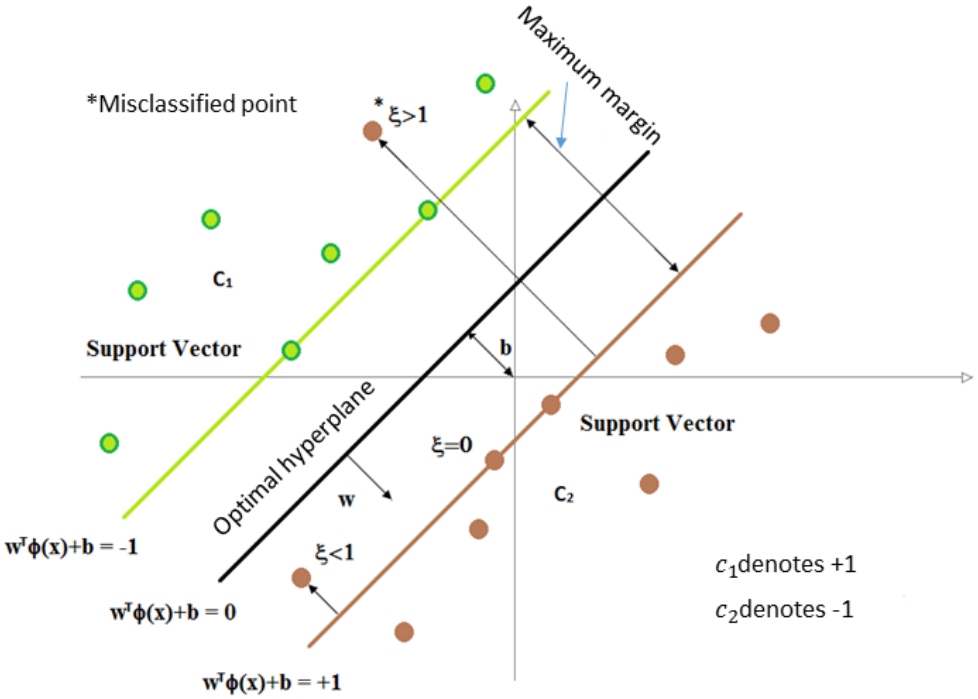


Figure 10. *Margin and the optimal separating hyperplane in SVM.*

To understand the basic theoretical foundations of support vector machines to linearly separate two-class data, in the binary classification problems, a set of training samples $\{x_i, i = 1, .., n\}$ are allocated to one of two classes, C_1 and C_2 , with corresponding labels y_i = ±1.

Let us define the linear discriminant function $g(x)$ as

$$g(x) = w^T x + w_0 \tag{1}$$

with the decision rule

$$\begin{cases} w^T x + w_0 > 0 \\ w^T x + w_0 < 0 \end{cases} \Rightarrow x \in \begin{cases} C_1, & y_i = +1 \\ C_2, & y_i = -1 \end{cases}.$$

Therefore, if $y_i(w^T x + w_0) > 0$, for all *i*, all training points can be perfectly classified.

If a perceptron rule is introduced to obtain a margin, $b > 0$, then it can be written as,

$$y_i(w^T x_i + w_0) \geq b, \tag{2}$$

where x_i are located at a distance greater than $b/|w|$ from the separating hyperplane. By determining a scaling of w and b, w_0 keeps this distance unchanged and condition (2) is still satisfied.

In the derivation of nonlinear SVM, a basic knowledge of mapping functions called kernels is introduced. The rule of kernel functions in support vector machines is to provide a scalar measure of similarity between two sample instances. A kernel function is defined by:

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle , \ \Phi: X \to H. \tag{3}$$

Here, *x* and *x'* are instances in sample space X. The dot product between sample vectors is one example of a kernel function, which is defined in equation (3). The kernel function $k(x, x')$ maps these instances into the feature space H by using the mapping Φ and presents the similarity in a proper form that makes it computationally feasible [86]

$$k(x, x') = \langle x, x' \rangle = \sum_{j=1}^{n} x_j x'_j. \tag{4}$$

It was mentioned that the general idea of SVMs is to map data into higher dimension to make it linearly separable. From equation (3), one can observe that this mapping is done by kernels using the mapping function Φ . The mapping could also be done explicitly to the data and only after that apply, e.g., the dot product. When the combined kernel is used, it is not necessarily needed to know the mapping function. Some commonly used kernels, in addition to linear kernel presented in equation (4), are the polynomial kernel of order *p* using constant term *d*,

$$k(x, x') = (\langle x, x' \rangle + d)^p, \tag{5}$$

the Gaussian radial basis function (RBF) kernel,

$$k(x, x') = e^{\left[\frac{\|x - x'\|^2}{2\sigma^2}\right]} \tag{6}$$

and the sigmoid kernel from neural networks,

$$k(x, x') = \tanh(k\langle x, x'\rangle + \Theta). \tag{7}$$

All nonlinear kernels here have tunable parameters. Tuning these parameters is one of the problems in designing and training an SVM classifier. E.g., in the RBF kernel the width of the kernel affects how well the classifier performs. Large widths cause the classifier to become too general and nonlinear mapping is no longer advantageous. Meanwhile, if the kernel is too narrow, the classifier over fits and its performance is poor.

Soft margin radial basis function SVM classifiers are defined here as in [71], [83]. In the linear model one assumes that the data is linearly separable and that there is a dot product space H and sample vectors $x_1, \dots, x_m \in H$. A hyperplane in H can in that case be defined as:

$$\langle x, w\rangle + b = 0 \tag{8}$$

where w is orthogonal to the hyperplane.

The location of the sample with respect to the hyperplane is then what determines the decision of the classifier, as follows:

$$f(x) = \operatorname{sgn}(\langle x, w\rangle + b) \tag{9}$$

where

$$\operatorname{sgn}(l) = \begin{cases} 1 & l > 0 \\ 0 & l = 0 \\ -1 & l < 0 \end{cases}.$$

Adding margins to the linear classifier generates the idea of support vector machines, from the maximization of this margin between the classes. Let the margin be:

$$\langle x_i, w\rangle + b \geq 1, y_i = +1$$

$$\langle x_i, w\rangle + b \leq -1, y_i = -1$$

or

$$y_i(\langle x_i, \boldsymbol{w}\rangle + b) \geq 1 \quad \forall i. \tag{10}$$

From this, the weight vector defines the margin which is $1/\|\boldsymbol{w}\|$. Meanwhile, the minimum distance between two points from different classes is $2/\|\boldsymbol{w}\|$. To find the weight vector $\boldsymbol{w}$ of the maximal margin, one can construct an optimal hyperplane by:

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{subject to } y_i(\langle \boldsymbol{x}_i, \boldsymbol{w}\rangle + b) \geq 1 \quad \forall i. \tag{11}$$

This equation defines an optimization problem. To deal with it we convert it into a dual optimization problem for which we can derive a solution, by introducing the Lagrangian:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(\langle \boldsymbol{x}_i, \boldsymbol{w}\rangle + b) - 1\right), \tag{12}$$

where the Lagrange multipliers $\alpha_i \geq 0$.

This Lagrangian L must be maximized with respect to $\boldsymbol{\alpha}$ and minimized with respect to $\boldsymbol{w}$ and b. In order to eliminate the derivatives of L with respect to $\boldsymbol{w}$ and b, we have two solutions:

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{13}$$

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i. \tag{14}$$

This optimization problem has optimality conditions, defined as Karush-Kuhn-Tucker (KKT) conditions. According to the KKT theorem, the Lagrange multipliers α_i greater than zero provide the solution [71], [83]. From equation (14) the solution vector can be expanded based on the training samples. Those training samples for which $\alpha_i \geq 0$ are the 'support vectors'.

Next, equations (13) and (14) must be introduced into the Lagrangian, equation (12), as follows:

$$\begin{gathered} \max W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ \text{subject to } \alpha_i \geq 0, i = 1, \ldots, m \text{ and } \textstyle\sum_{i=1}^{m} \alpha_i\, y_i = 0. \end{gathered} \tag{15}$$

When equation (14) is introduced into equation (9) one gets:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b\right). \tag{16}$$

This equation can be evaluated using the dot products between sample to be classified and support vectors. The support vectors are the only training samples needed (instead of the entire training set), reducing the required computation.

If the assumption of separability is invalid, one can use the kernel functions to map the data into a feature space, where the classes are linearly separable.

The non-linear SVM is obtained from the linear one by this mapping as follows:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b\right). \tag{17}$$

Equation (17) is obtained when the dot product in equation (16) is replaced with kernel function.

Not all nonlinear problems are separable and can be solved with linear classifier after this mapping. Soft margin SVM is a technique to address this problem. It accepts some errors in the classification, i.e. some samples are left out when determining the hyperplane. To allow this the margin must be redefined as:

$$y_i(\langle \boldsymbol{x}_i, \boldsymbol{w} \rangle + b) \geq 1 - \xi_i \quad \forall i, \xi \geq 0. \tag{18}$$

In this equation, ξ_i are the 'slack' variables. From this equation, if ξ_i are large enough the constraints can be met. A form of penalization must be introduced to avoid a solution where all ξ_i are large. For this, we can add a term $\sum_i \xi_i$ to equation (11), thus getting:

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m} \xi_i \quad \text{subject to } y_i(\langle \boldsymbol{x}_i, \boldsymbol{w} \rangle + b) \geq 1 - \xi_i \quad \forall i, \\ \xi \geq 0 \tag{19}$$

where the constant C is a positive constant that constraints the second term of the equation, and determines the trade-off between minimizing the training error and maximizing the margin. Classification using soft margin SVM is done identically when using hard margin classifiers.

Now, once again the dual form of the problem becomes more easily treatable for finding a solution. Its derivation is the same as above. The coefficients are now found by solving:

$$\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ \text{subject to } 0 \geq \alpha_i \leq \frac{C}{m}, i = 1, \dots, m \text{ and } \textstyle\sum_{i=1}^{m} \alpha_i \, y_i = 0. \tag{20}$$

Unfortunately, there are no means to select the value of this additional tunable parameter a priori. Also, the errors introduced in the classification on purpose can affect the classifier's performance. Although increasing the computational complexity, a soft margin SVM has been found [87] that outperforms other SVM implementations.

SVMs can also be used for multiclass problems if one constructs linear discriminant functions, defined by:

$$\boldsymbol{g_k}(\boldsymbol{x}) = (\boldsymbol{w}^k)^T\boldsymbol{x} + \ w_0^k \ \ k = 1, \dots, C. \tag{21}$$

The goal is to find a solution for $\{(\boldsymbol{w}^k, w_0^k), k = 1, \dots, C\}$ such that the training data are separated without error, by determining the decision rule:

if $g_j(\boldsymbol{x}) = \max_j g_j(\boldsymbol{x})$, assign *x* to class w_i.

For $\{(\boldsymbol{w}^k, w_0^k), k = 1, \dots, C\}$, there are solutions such that,

$(\boldsymbol{w}^k)^T\boldsymbol{x} + \ w_0^k - ((\boldsymbol{w}^j)^T\boldsymbol{x} + \ w_0^j) \geq 1$ for all $k = 1, \dots, C$, all $\boldsymbol{x} \in \omega_k$, and all $j \neq k$,

which makes each pair of classes separable [53].

Here, to apply SVM as a multiclass classifier, we used the binary classifier in a one-against-all (OAA) procedure [53]. For that we built C binary classifiers for the C classes. The k^{th} classifier was trained to distinguish the samples in class w_k from all other ones. The weight vector w^k and the threshold, w_0^k, can be defined as:

$$(\boldsymbol{w}^k)^T\boldsymbol{x} + \ w_0^k \begin{cases} > \ 0 \\ < \ 0 \end{cases} \Rightarrow \boldsymbol{x} \in \begin{cases} w_k \\ w_1, \dots, w_{k-1}, w_{k+1}, \dots, w_C \end{cases}$$

For a sample *x*, $g_k(\boldsymbol{x}) = (\boldsymbol{w}^k)^T\boldsymbol{x} + w_0^k$ will be positive for a value of *k* and negative for the rest so as to most ideally define a class. However, it is common that a pattern *x* is classified into more than one class, or into none.

Some solutions to this have been proposed. For more than one selected class, *x* may be assigned only to the class for which $((\boldsymbol{w^k})^T\boldsymbol{x} + \boldsymbol{w}_0^k)/|\boldsymbol{w}^k|$ is the largest. Meanwhile, no class is selected, *x* can be assigned to the class with the smallest $((\boldsymbol{w^k})^T\boldsymbol{x} + \boldsymbol{w}_0^k)/|\boldsymbol{w}^k|$ [53].

We also defined the radial basis function kernel as a mapping function. To select a proper model, two parameters were determined, the 'rbf_σ' scaling factor of the kernel and the soft margin (C). These parameters were estimated by using cross-validation [88]. Figure 11 shows the block diagram of the SVM method used in this research.

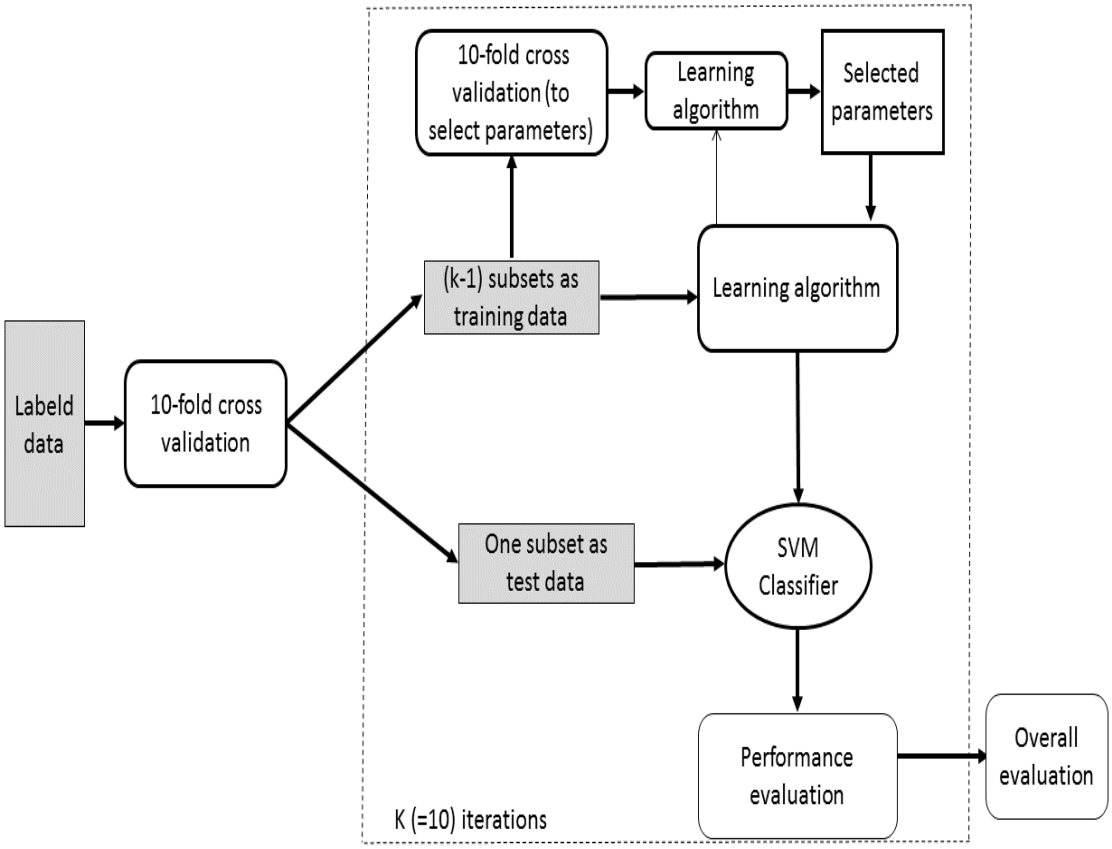


Figure 11. *Block diagram of Support Vector Machine method (SVM).*

3.7.3 Regularized Multinomial Logistic Regression

Logistic regression (LR) [89] is a statistical method. This method describes the relationship between a binary response variable and several predictor variables, called covariates.

In classification problems, LR can be used as a linear classifier. In this case, one can predict the class variable C through the covariates, $X_1, \dots, X_k$. This classifier can be utilized as a strong supervised classification model to provide the probability of classification, which is used to obtain class label information.

For two-class classification problems by applying LR classifier, The classification process can be carried out as follows; given a training data set D_N containing *N* independent samples $D_N = \{(c_j, x_{j1}, \dots, x_{jk}), j = 1, \dots, N\}$ by the joint probability distribution on (C, $X_1, \dots, X_k$), class label C can only take 0 and 1 values. If the label c_j = 1, the j^{th} input instance $x_j = (x_{j1}, \dots, x_{jk})$, which has the feature that C provides, belongs to the first class. If c_j = 0, x_j belongs to the other class. The class label of a new instance is determine by using the classification model [90].

LR models rely on the logistic sigmoid function. Given a class C_1, the posterior probability of this class can be written as follows [90]:

$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)} = \frac{1}{1 + ex\,p(-\alpha)} = \sigma(\alpha), \quad (22)$$

where α is defined as:

$$\alpha = \ln \frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)} \quad (23)$$

and $\boldsymbol{\sigma(\alpha)}$ is the logistic sigmoid function defined by

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)} \ . \quad (24)$$

The $\sigma(\alpha)$ is shown in Figure 12.

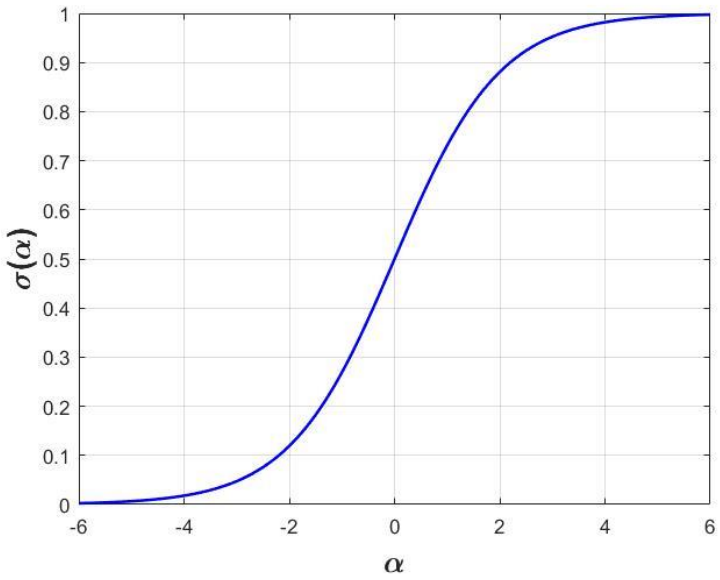


Figure 12. *Logistic sigmoid function.*

Since the posterior probability of class C_1 can be defined as a logistic sigmoid performing on a linear function of the feature vector ϕ, then

$$p(C_1|\phi) = y(\phi) = \sigma(w^T\phi), \quad (25)$$

where *w* is a weight vector. And, in the following, we have $p(C_2|\phi) = 1 - p(C_1|\phi)$. In the terms of statistics, this model is regarded as logistic regression for classification [88].

When we deal with an *M*-dimensional feature space φ, the LR model would have *M* parameters that should be adjusted. On the other hand, if one uses the maximum likelihood to fit the Gaussian class conditional, there would be a large number of parameters, 2M parameters arise from means and *M* (*M* + 1)/2 parameters correspond to the covariance

matrix. Therefore, the total number of parameters, by considering the class prior $p(C_1)$, would be M (M + 5)/2 + 1. I.e., the number of parameters to be adjusted quadratically increases with M.

For large values of M, the logistic regression model can be applied directly, and the maximum likelihood can be used to calculate its parameters [88]. To perform this, the derivative of the logistic sigmoid function is introduced. This function can be expressed in terms of the sigmoid function [88], which is written as,

$$\frac{d\sigma}{d\alpha} = \sigma(1-\sigma)\,. \tag{26}$$

For a data set $\{\phi_n, t_n\}$ in which $t_n \in \{0, 1\}$ are class labels and $\phi_n = \phi(x_n)$ are feature vectors, with $n = 1, \dots, N$, the likelihood function can be defined as:

$$p(t|w) = \prod_{n=1}^{N} y_n^{t_n} \{1-y_n\}^{1-t_n}, \tag{27}$$

where $t = (t_1, \dots, t_N)^T$ and $y_n = p(C_1|\phi_n)$.

From the negative logarithm of the likelihood, the cross-entropy error function can be defined as:

$$E(w) = -\ln p(\boldsymbol{t}|w) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}, \tag{28}$$

where $y_n = \sigma(\alpha_n)$ and $\alpha_n = w^T\phi_n$. The gradient of the error function with respect to w can be computed as,

$$\frac{\partial E}{\partial y_n} = \frac{1-t_n}{1-y_n} - \frac{t_n}{y_n} = \frac{y_n(1-t_n) - t_n(1-y_n)}{y_n(1-y_n)} = \frac{y_n - t_n}{y_n(1-y_n)}. \tag{29}$$

From equation (26), we have

$$\frac{\partial y_n}{\partial \alpha_n} = \frac{\partial \sigma(\alpha_n)}{\partial \alpha_n} = \sigma(\alpha_n)(1-\sigma(\alpha_n)) = y_n(1-y_n). \tag{30}$$

At the end, by defining ∇ as the gradient with respect to w, we have

$$\nabla \alpha_n = \phi_n. \tag{31}$$

After combining equation (29), (30), and (31), the chain rule is applied to obtain

$$\nabla E(w) = \sum_{n=1}^{N} \frac{\partial E}{\partial y_n} \frac{\partial y_n}{\partial \alpha_n} \nabla \alpha_n = \sum_{n=1}^{N} (y_n - t_n)\, \phi_n, \tag{32}$$

where the ϕ_n is basis function vector and $(y_n - t_n)$ is the error between the target value and the prediction of the model. Note that the factor in the equation (26) has been removed, therefore simplifying the gradient of the log likelihood.

It is worth mentioning that, for data sets that are linearly separable, the maximum likelihood can lead to severe over-fitting. The main reason is that when the hyperplane corresponding to σ = 0.5 reaches zero, the maximum likelihood solution occurs when $w^T\phi = 0$, and the magnitude of *w* goes to infinity. This is called a singularity [88].

It should be noticed that even when the number of data points is larger than the number of the model parameters, the problem will occur, as long as the training data set can be separated linearly. To prevent the singularity, the solution can be incorporating a regularization term to the error function, or one can include a prior and determine a maximum a posteriori (MAP) solution for *w*.

For logistic regression, because of the nonlinearity property of the logistic sigmoid function, there is no a closed-form solution. Furthermore, we can apply an iterative technique based on the Newton-Raphson iterative optimization measure to minimize the error function. In this approach, a local quadratic approximation is used in the log likelihood function. To minimize the function *E(w)*, the update of Newton-Raphson comes in the form of [88]:

$$w^{(new)} = w^{old} - \boldsymbol{H}^{-1} \nabla \mathrm{E}(w), \tag{33}$$

where **H** is the Hessian matrix. The elements of this matrix are the second derivatives of E(*w*) with respect to the components of *w*.

The posterior probabilities for multiclass classification models, for a large class of distributions, are computed through a softmax transformation of linear functions of the feature variables, so that:

$$p(C_k|\phi) = y_k(\phi) = \frac{\exp(\alpha_k)}{\sum_j \exp(\alpha_j)}, \tag{34}$$

where *k* is the number of classes and α_k are given by

$$\alpha_k = \mathrm{w}_k^T \phi\ .$$

The maximum likelihood is used to directly calculate the parameters $\{w_k\}$ of this model. Furthermore, the derivatives of y_k with respect to all of the activations α_j will be required. These are computed as:

from equation (34), we see that

$$\frac{\partial y_k}{\partial \alpha_k} = \frac{\exp(\alpha_k)}{\sum_j \exp(\alpha_j)} - \left(\frac{\exp(\alpha_k)}{\sum_j \exp(\alpha_j)}\right)^2 = y_k(1 - y_k),$$

$$\frac{\partial y_k}{\partial \alpha_j} = -\frac{\exp(\alpha_k)\exp(\alpha_j)}{\left(\sum_j \exp(\alpha_j)\right)^2} = -y_k y_j, \quad j \neq k,$$

Therefore,

$$\frac{\partial y_k}{\partial \alpha_j} = y_k(I_{kj} - y_j),$$

where I_{kj} are the elements of the identity matrix.

The likelihood function is then defined by

$$p(T|w_1, \dots, w_K) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(C_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}, \tag{35}$$

where $y_{nk} = y_k(\phi_n)$, and T is an $N \times K$ matrix of target variables with elements t_{nk}.

The negative logarithm is then written as

$$E(w_1, \dots, w_K) = -\ln p(T|w_1, \dots, w_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}. \tag{36}$$

This is known as the cross-entropy error function for the multiclass classification problem [88]. Next, the gradient of the error function is taken with respect to one of the parameter vectors w_j:

$$\nabla_{w_j} E(w_1, \dots, w_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\, \phi_n$$

where we have made use of $\sum_k t_{nk} = 1$.

To find a batch algorithm, the Newton-Raphson update is again applied. To do this, it is required to evaluate the Hessian matrix that consists of blocks of size $M \times M$ in which the block *j*, *k* is given by

$$\nabla_{w_k}\nabla_{w_j} E(w_1, \dots, w_K) = \sum_{n=1}^{N} y_{nk}(I_{kj} - y_{nj})\, \phi_n \phi_n^T . \tag{37}$$

Like the two-class problem, the Hessian matrix for the multiclass logistic regression model leads a unique minimum for the error function [88]. In this study, we used the regularized multinomial LR method. This method generalizes LR to multiclass problems [91]. We used this method as implemented in the MATLAB^{TM} package (https://se.mathworks.com/matlabcentral/fileexchange/55863-logistic-regression-for-classification). In this implementation, in order to prevent the singularity problem that was mentioned above, a regularization parameter (λ) is defined. The default value of this parameter is determined as 10^{-4}. In this method, the parameters are estimated using the maximum likelihood principle and Newton-Raphson algorithm is also used to solve the likelihood equations numerically. Figure 13 shows the block diagram of the used RML classifier. The same as DT method, we also used 10-fold cross-validation technique to split the data so that can be used to learn the model and then use the learned model to classify the new instance.

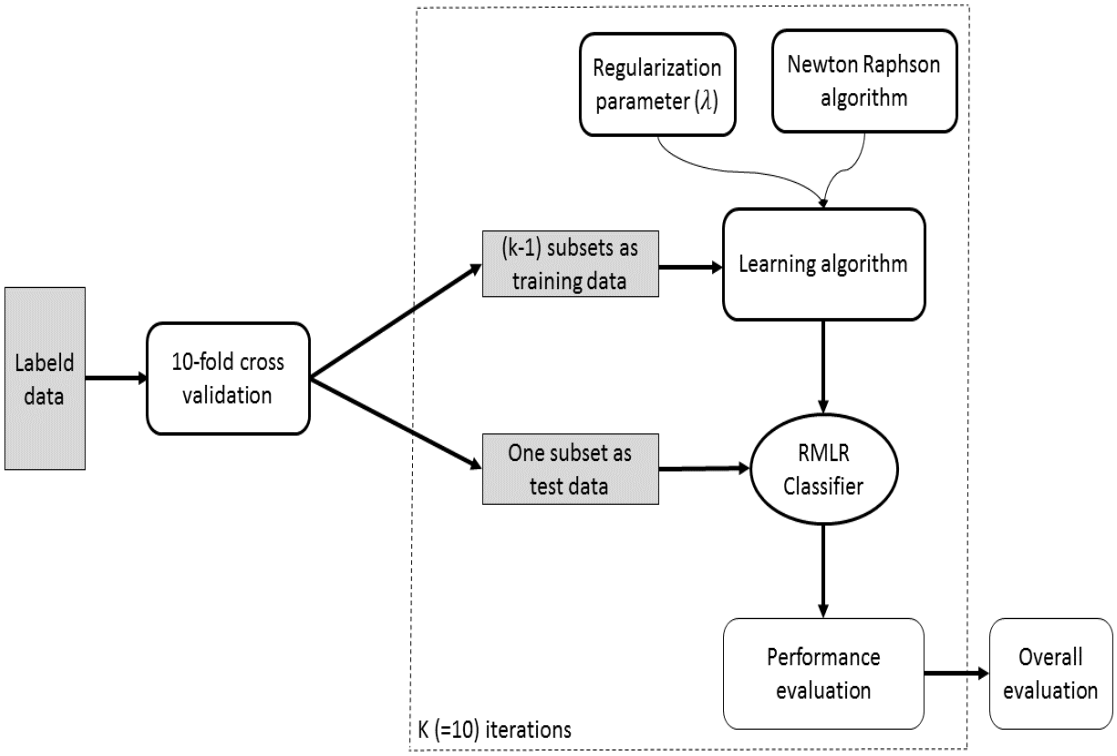


Figure 13. *Regularized Multinomial Logistic Regression (RMLR) classifier scheme.*

3.8 Methods of classifier performance evaluation

A simple measure of quality of a classifier is that it can provide prefect classification results. The evaluation of the classification results is taken as an important part of the classifier design process and is usually used as a basis for selecting the more proper classifier for the problem in question. It can be turned into error estimation to produce an estimate of the errors made by a chosen classifier. Some performance evaluation methods are presented in following text. Note that, all these methods are naturally heuristic and

we would not be able to present the most reliable method to test which classifier generalizes best on an arbitrary problem.

A very simple method to evaluate the performance of a classifier is to measure the classification rate or accuracy (ACC), which is measured by dividing the number of correctly classified samples by the total number of samples. However, in some problems, the ACC alone typically does not provide enough information to allow determining the efficiency of the classifier. For example, when we have a model that we believe that it can make robust predictions, we need to decide whether it is a good enough model to solve our problem. Therefore, we need more information to make this decision.

Confusion matrix is another technique that is often used to describe the performance of a classification model. A confusion matrix [92] contains information about actual and predicted classifications obtained by a classification system. The data in the matrix is commonly used to evaluate the performance of such systems. The following table, Table 1, shows a confusion matrix for a three class classifier.

Table 1. *Confusion Matrix for three classes.*

Actual class	**Predicted class**		
C_1	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$
C_2	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$
C_3	$c_{3,1}$	$c_{3,2}$	$c_{3,3}$
Column totals	n_1	n_2	n_3

The ACC of a multiclass classifier, for example in the case of three classes, can also be computed from a confusion matrix, as is defined by

$$ACC = \frac{c_{1,1} + c_{2,2} + c_{3,3}}{n_1 + n_2 + n_3}. \tag{38}$$

By the above equation, we can split Table 1 into three separate tables, using the one-against-all technique. Each time, we can consider one of the classes as the positive class and the other two as the negative classes. Therefore, we have three tables as exemplified in Table 2.

Table 2. *Confusion matrix for two classes.*

Actual class	**Predicted class**	
Positive	True Positive (TP)	False Positive (FP)
Negative	True Negative (TN)	False Negative (FN)

For example, if we consider C_1 as the positive class, TP, FP, TN, and FN are computed as:

- TP: the number of C_1 samples that are classified as C_1

- FP: the number of C_2 and C_3 samples that are classified as C_1
- TN: the number of C_2 and C_3 samples that are not classified as C_1
- FN: the number of C_1 samples classified that are not classified as C_1.

Based on the applications, confusion matrices can be used to obtain different performance measures. For example, sensitivity and specificity are statistical measures that can be computed by a confusion matrix. The sensitivity (SEN), or true positive rate, measures the percentage of positive labelled samples that are correctly classified as positive. High sensitivity values imply that there are few false negatives. The specificity (SPE), or true negative rate, measures the percentage of negative labelled samples that are correctly classified as negative. High specificity values imply that there are few false positive results. These quantities are computed as follows,

$$SEN = \frac{TP}{TP + FN} \tag{39}$$

$$SPE = \frac{TN}{TN + FP}. \tag{40}$$

The receiver operating characteristic (ROC) curve, a list of ROC points from (0,0) to (1,1), is another performance measure, which plots the true positive rate or sensitivity, on the vertical axis, against the false positive rate or costs, on the horizontal axis, of a classifier. The false positive rate (FPR) equation can be written as,

$$FPR = 1 - SPE = \frac{FP}{FP + TN}. \tag{41}$$

In this research, we computed SEN and SPE of each class by using the confusion matrix in Table 2 and the one-against-all technique, described above.

ROC can also be regarded as a good means of visualizing a classifier's performance to select a suitable decision threshold. In practice, like the error rate, the optimal ROC curve (obtained from the true class-conditional densities, $p(x|\omega_i)$, where ω_i denotes classes) is uncertain and must be estimated using a trained classifier and an independent test set of samples with known labels. Nevertheless, as for the error rate estimation, a training set reuse method such as cross-validation can be used as well [53].

One method that we used to handle *n* classes is to produce *n* different ROC graphs, one for each class, called the *class reference* formulation. If we define C as the set of all classes, ROC curve *i* plots the classification performance by taking class c_i as the positive class and all other classes as the negative class [93], defined as

$$P_i = c_i$$

$$N_i = \bigcup_{j \neq i} c_j \in C.$$

Area under ROC (AUC) [94] is computed to evaluate the overall classification performance. The advantage of the AUC is that it is a ranking-based performance measure. Its value can be regarded as the probability that a classifier is able to discriminate between a randomly chosen positive example and a randomly chosen negative example. Compared with many alternative performance measures, AUC is invariant to class-specific error costs and relative class distributions [95].

The AUC in a two-class problem is a single scalar value [96]. For the AUC value in a multi-class problem, the issue of integrating values of multiple pairwise discriminability is introduced. One technique to compute multi-class AUCs was proposed by Provost and Domingos in 2000 [97] . In this approach, to calculate AUCs for multi-class problems, one can provide each class reference ROC curve, measure the area under the curve, and then sum the AUCs, which are weighted by the reference class's prevalence in the data [93], which is defined by

$$\boldsymbol{AUC_{total}} = \sum_{\boldsymbol{c_i \in C}} \boldsymbol{AUC}(\boldsymbol{c_i})\boldsymbol{p}(\boldsymbol{c_i}). \tag{42}$$

A realistic classifier should obtain an AUC higher than 0.5 [93].

Another technique to achieve proper evaluation results is to divide our samples into three distinct sets. One set is used for training the model, or classifier, the second set is used to select the best trained model, and the third set, named the test set, is used to estimate the performance of the selected model.

In some classification problems with enough samples, it is possible to use this simple method. Commonly, the data is randomly divided so that half of the data is used as a training set while the test set and the validation set cover one fourth of the data, each [75]. In many data-poor situations, the simple partition method described is not efficient. Instead, one can use Resubstitution methods. In these, while the validation set is randomly selected from the data, the entire dataset is still used for training. The disadvantage of resubstitution methods is that they tend to present excessively positive estimates for the classification error. The cross validation (CV) [52] is one the methods that use this kind of resampling with better error estimates, is described in Section 3.6.4.

4. RESULTS

In this Thesis, we studied classification methods of images applied to the problem of automatically classification of the stages of Z-ring formation in individual cells. The true class of Z-ring formation were defined by the biologists.

In order to compare the efficiency of the various classification methods used in determining the stage of formation of each Z-ring from fluorescence microscopy images, we first collected images of cells by phase contrast (to detect the cell borders) as well as by confocal microscopy (to visualize fluorescently tagged FtsZ proteins) (Chapter 3). Next, using *CellAging* [29] and MAMLE [66], we performed image analysis for cell segmentation and labelling.

Afterwards, from the images, 50 cells (labelled samples) were randomly collected as representative of each of the three classes by an expert, for a total of 150 labelled samples.

From the distribution of fluorescence intensity from FtsZ-GFP along the major cell axis of each cell, we extracted the mean, variance, standard deviation (std), skewness, and first order histogram. From these features, by pre-analysis, we selected the std and the mean as the features for classification, as they exhibited more ability to best distinguish the stages of FtsZ ring formation when compared to the other features tested.

Table 3 shows labelled data. Each row and column in the table present each class (C_1, C_2, and C_3) and the measured features from the sample, namely, the std and mean of fluorescence intensity of each section of the cell (as described in section 3.6.2).

Visibly, the maximum cloud-like distribution of fluorescence intensity of cells from class C_1 is located in one of the poles. While, in cells from the two other classes, C_2 and C_3, maximum cloud-like distributions are located at the center of the cell, which is shown by gray color in Table 3. There is also one additional difference that allows to distinguish between the classes C_2 and C_3: the intensity in cells from C_3 is higher. Therefore, this feature can be exploited to efficiently discriminate C_2 from C_3 cells.

Table 3. *Examples of labelled data, in which there are 5 samples in each class. the gray color shows the cloud-like distribution of fluorescence intensity of cells.*

	features						
classes	left pole		midcell		Right pole		labels
	Std	mean	std	mean	std	mean	
C_1	0,1239	0,1055	0,0532	0,0704	0,0229	0,0185	1
C_1	0,2774	0,2457	0,0333	0,0391	0,0102	0,0117	1
C_1	0,0147	0,0221	0,0230	0,0299	0,2112	0,1936	1
C_1	0,0876	0,1127	0,1455	0,1741	0,2659	0,3369	1
C_1	0,0310	0,0313	0,0300	0,0464	0,1793	0,1442	1
C_2	0,0352	0,0311	0,1841	0,1743	0,0211	0,0264	2
C_2	0,0435	0,0557	0,1643	0,1644	0,0350	0,0396	2
C_2	0,0320	0,0637	0,1918	0,1889	0,0229	0,0327	2
C_2	0,0550	0,0579	0,1896	0,1751	0,0263	0,0301	2
C_2	0,0429	0,0451	0,1913	0,1806	0,0278	0,0271	2
C_3	0,0396	0,0431	0,2168	0,2056	0,0713	0,0739	3
C_3	0,0249	0,0335	0,2139	0,1864	0,0218	0,0283	3
C_3	0,0546	0,0586	0,1851	0,2194	0,0355	0,0462	3
C_3	0,0339	0,0407	0,1571	0,1497	0,0258	0,0254	3
C_3	0,0455	0,0616	0,1803	0,2089	0,0519	0,0933	3

To each of these three classes of cells, we subsequently applied three classification methods (see Chapter 3). The performance of each method was estimated by four techniques. First, it estimated by the ACC or classification rate, which is the ratio of correctly classified samples to the total number of samples averaged over the folds. Also, we calculated the AUC, which is the classification performance measure based on the ROC curve (the results of this method are presented later in this section). Next, we calculated the SEN, which is the ratio of correctly classified positive samples to the true positive samples. Finally, we calculated the SPE, which is the ratio of correctly classified negative samples

to the true negative samples. The final results are averaged over a hundred 10-fold cross-validated (CV) runs, to minimize the effect of the random variation.

To avoid the multiclass classification problem [98], we started with a decision tree (DT). The ACC, SEN, and SPE of this method with 10-fold cross-validation were measured to be 67.33% , 67.28%, 83.65% , respectively (Table 4). However, we expect that this can be improved by simple techniques, such as pruning, which would remove parts of the tree that contribute little to the ability to classify the samples.

Table 4. *The performance of DT, SVM, and RMLR, using 150 labelled samples.*

Method	ACC (%)	AUC (%)	SEN (%)	SPE (%)
DT	67.33	54.37	67.28	83.65
SVM	72.19	61.97	71.96	86.03
RMLR	77.34	72.13	77.91	88.80

We also computed the confusion matrix for the DT classifier, shown in Table 5. It can be seen that 48 out 50 cells from class C_1 were correctly classified to class C_1, namely, they were correctly identified as being in the first stage, while the other 2 cells were misclassified to C_3. Meanwhile, 24 out of 50 cells from class C_2 were correctly classified to class C_2, 2 was misclassified to class C_1, and the remaining ones were misclassified to class C_3. Finally, 28 out of 50 cells from class C_3 were misclassified to class C_2, with the remaining ones being accurately classified.

Table 5. *Confusion matrix of DT classifier.*

Actual class	**Predicted class**		
Class_1	48	2	0
Class_2	0	24	28
Class_3	2	24	22
Column totals	50	50	50

We also plotted the ROC curve of classes (Figure 14), one ROC curve is drawn per class, to evaluate the ability of DT classifier to discriminate the classes. From Figure 14, we can conclude that most cells from class C_1 could be correctly detected and classified, with low error rate, compared with the cells from classes C_2 and C_3. The total AUC of this classifier was also computed using equation (42), equation AUC_{total}, and Table 5. From this, the AUC was found to be 54.37% (Table 4).

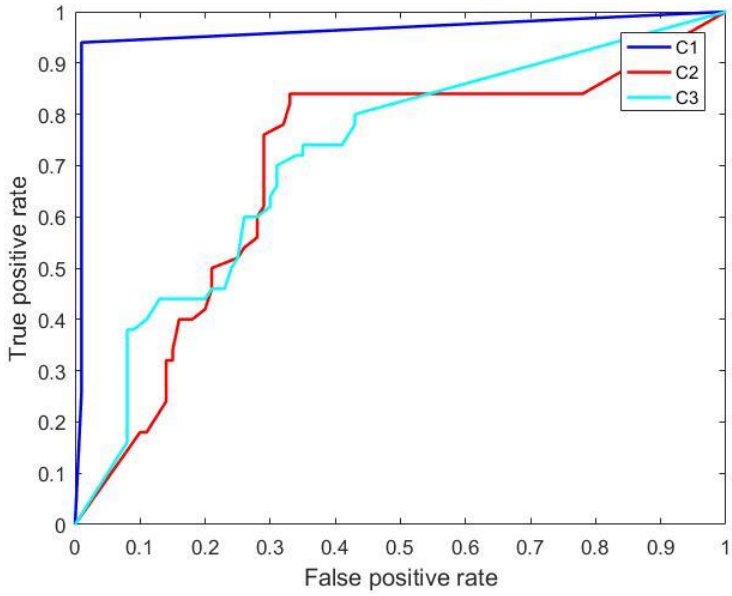


Figure 14. *ROC curves of DT classifier, one ROC curve is shown per class.*

Next, we applied the SVM classifier. To enhance the ACC, we determined two parameters for SVM, namely, the 'rbf_σ' scaling factor of the radial basis function kernel and the 'boxconstraint' (C), a soft margin parameter. The parameters were adjusted by using a grid search, that selected amongst the following candidate sets: $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ for C and $\{10^{-1}, 2\times10^{-1}, 4\times10^{-1}, 6\times10^{-1}, 8\times10^{-1}, 10^{0}\}$ for rbf_σ.

In order to evaluate each possible combination of these parameters, a cross-validation technique (described in Section 3.6.4) is used, however, this increased the time needed to implement the method. The ACC, SEN, and SPE are shown in Table 4. The confusion matrix for this classifier is in Table 6. The output of SVM classifier shows that the performance of this classifier is better than DT, since, in total, only 43 out of 150 cells are misclassified.

Table 6. *Confusion matrix for SVM classifier.*

Actual class	**Predicted class**		
Class_1	49	0	0
Class_2	0	31	23
Class_3	1	19	27
Column totals	50	50	50

The ROC curve of each class, from SVM classifier, is shown in Figure 15. It can be seen that, the same as DT classifier, this classifier could also detect and classify more cells from class C_1 correctly. However, SVM, totally, outperforms DT, the AUC_{total} of this classifier was measured to be 61.97% (Table 4).

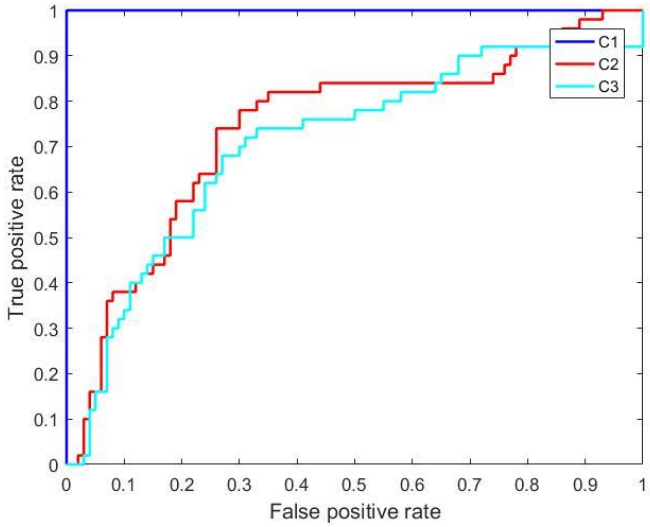


Figure 15. *ROC curves of SVM classifier, one ROC curve is shown per class.*

Finally, we used the RMLR classifier as implemented in the MATLAB^{TM} PRML toolbox (https://se.mathworks.com/matlabcentral/fileexchange/55863-logistic-regression-for-classification). We used the default value of this classifier, regularization parameter (λ), 10^{-4}. Even without tuning this parameter, we obtained a 10-fold cross-validated ACC of 77.34%, SEN of 77.91%, SPE of 88.80% (Table 4).

The measured confusion matrix of RMLR is in Table 7. The number of misclassified cells by this method is lower than by the DT and the SVM methods, with only 30 out of 150 cells being misclassified.

Table 7. *Confusion matrix for RMLR.*

Actual class	**Predicted class**		
Class_1	49	0	0
Class_2	1	34	13
Class_3	0	16	37
Column totals	50	50	50

The ROC curve of each class, from the RMLR classifier, is shown in Figure 16. The AUC_{total} of this classifier was measured to be 72.13% (Table 4). The ROC curves of classes show that this classifier can distinguish better the classes than the previous methods.

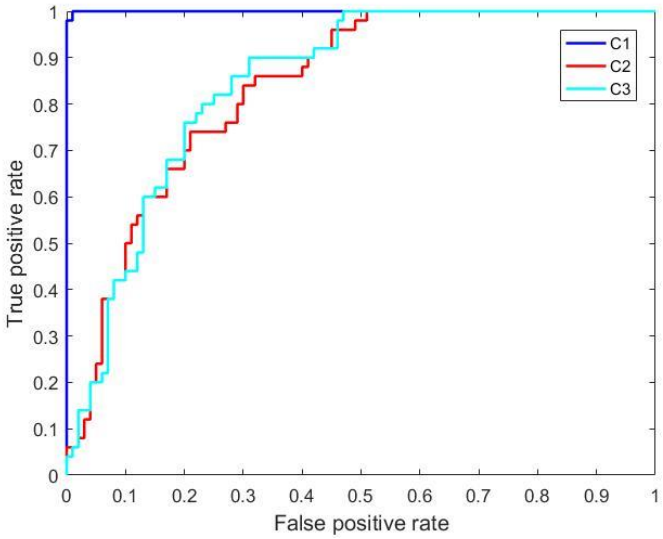


Figure 16. *ROC curves of RMLR classifier, one ROC curve is shown per class.*

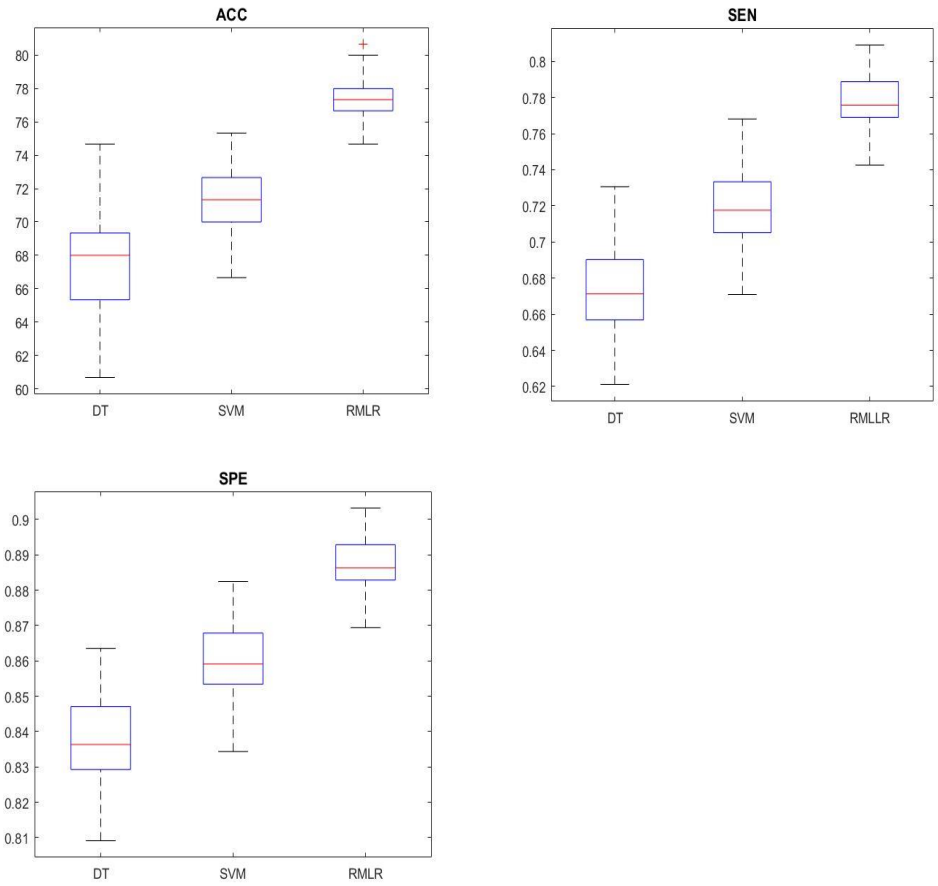


Figure 17. *Box plots of ACC, SEN, and SPE for DT, SVM, RMLR.*

Next, we compared the performance of all three classifiers when using 150 labelled data. Results are presented in the form of a boxplot of their ACC, SEN, and SPE, in Figure 17. The results are obtained for 10-fold cross validation and 100 computation times. These

results confirm the trends previously identified. Namely, the RMLR outperforms the other two classifiers. However, we can significantly visualize the benefits of additional data for the SVM in this figure.

The presented results associated with 150 labelled data indicate that RMLR had better performance and that this classifier was able to discriminate all three classes from each other, particularly class C_1 from two other classes C_2 and C_3, which can be seen in Figure 16, with AUC 72.13% (Table 4). SVM even outperforms DT with AUC 61.97% (Table 4), while the DT classifier had poor performance with 54.37% AUC (Table 4).

Since we split the labelled data into two sets, training and test sets, it leads to train our model with less number of training data, which can affect the performance of the classifiers. As the next step, we increased the number of labelled data to 300, with 100 samples per class. We also averaged the results over 10-fold cross validation and 100 computation times.

From Table 8, we can see how well the performance of the classifiers improved by increasing the number of the labelled samples. It can also be visualized that the performance of the SVM classifier become more similar to that of RMLR. In short, RMLR continued to outperform the others, and the DT is still quite inefficient in the classification procedure. The poor performance of DT, for our data, can be because of its non-parametric approach.

Table 8. *The performance of DT, SVM, and RMLR, by using 300 labelled samples.*

Method	ACC (%)	AUC (%)	SEN (%)	SPE (%)
DT	73.00	61.81	72.99	86.51
SVM	76.93	70.78	78.37	89.14
RMLR	79.44	75.12	79.40	89.63

Next, in Table 9, we show the confusion matrices for DT, SVM, and RMLR after increasing the number of labelled samples. Compared to these matrices for less samples, we find that, in all cases, the results improved. The method whose results most improved are those of the RMLR classifier, which was able to correctly classify 241 out of 300 samples.

Table 9. *Tables, from left to right, display the Confusion matrix for DT, SVM, and RMLR classifiers.*

Actual class	Predicted class		
Class_1	97	2	1
Class_2	1	63	41
Class_3	2	35	58
Column totals	100	100	100

Actual class	Predicted class		
Class_1	98	2	0
Class_2	1	81	45
Class_3	1	17	55
Column totals	100	100	100

Actual class	Predicted class		
Class_1	98	1	0
Class_2	2	70	27
Class_3	0	29	73
Column totals	100	100	100

Figure 18 displays the ROC curve of DT classifier after increasing the number of labelled data. Although the performance of this classifier improved, it is still poor in discriminating classes, compared with the two other classifiers.

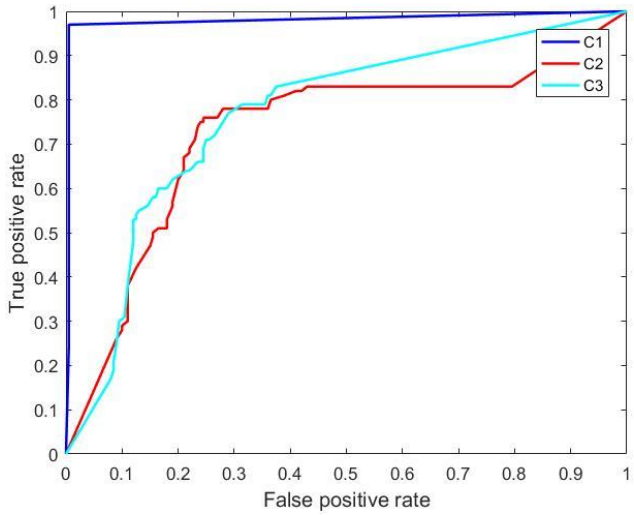


Figure 18. *ROC curve of DT, 300 samples.*

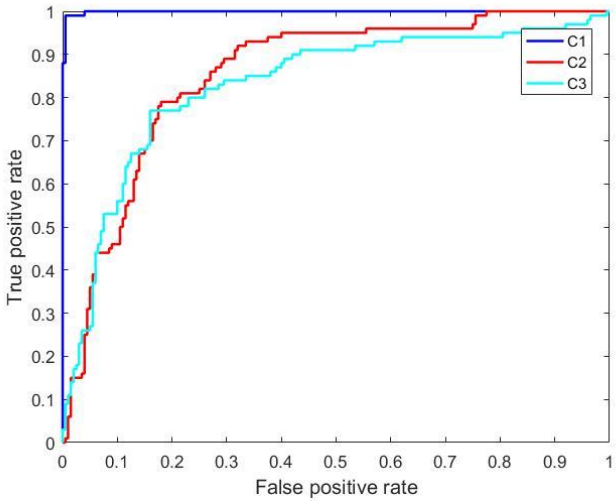


Figure 19. *ROC curve of SVM, 300 samples.*

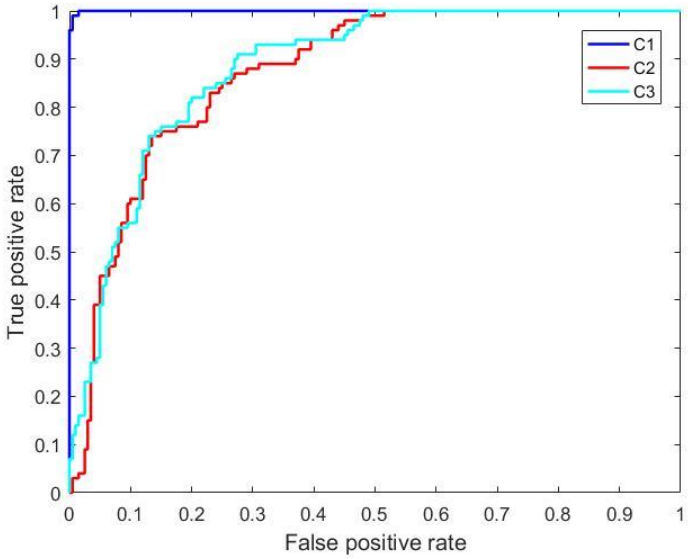


Figure 20. *ROC curve of RMLR, 300 samples.*

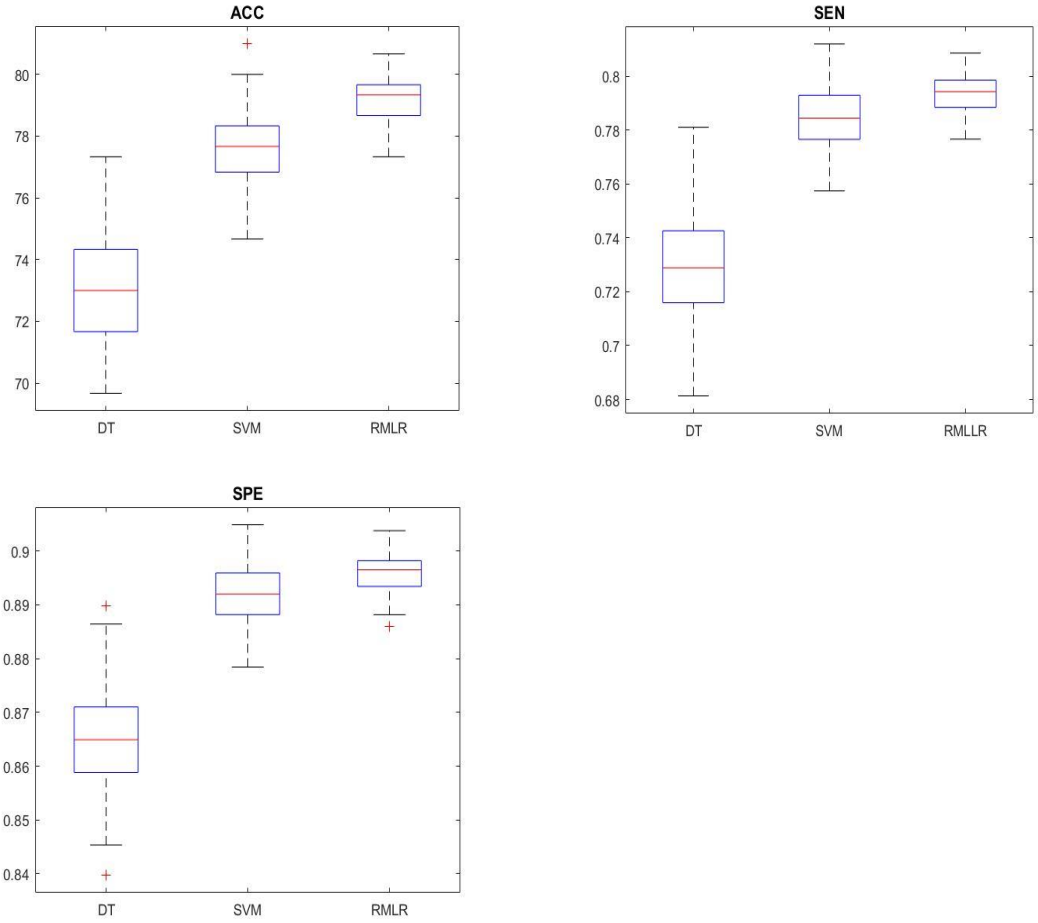


Figure 21. *Box plots of ACC, SEN, and SPE for DT, SVM, and RMLR, after increasing the number of labelled samples to 300.*

Figure 19 and Figure 20 illustrate the ROC curves of each class for SVM and RMLR, respectively. Meanwhile, in order to compare their performance, we plotted the box plots of ACC, SEN, and SPE of each classifier Figure 21.

Overall, from Table 9 and Figures 19-20, the classification methods produce far more errors when trying to distinguish between cells in stages 2 and 3 than between these two stages and stage 1.

5. DISCUSSION AND CONCLUSION

This thesis focused on the process of cell division in *E. coli* as observed by time-lapse microscopy of a modified strain with fluorescently tagged FtsZ proteins. Our aim was to automatically classify the stage of Z-ring formation of each cell from microscopy images by using image processing and machine learning methods. Recent developments in microscopy techniques as well as in fluorescent protein labelling techniques for observing internal cellular processes both in time and space have allowed a very rapid increase of knowledge on the dynamics of many cellular processes, from gene expression [99], [100] to cell division [2], [9] and cellular aging [101], [102].

Importantly, the developments observed in this research area commonly referred to as 'Single-cell Biology" would not have been possible without the parallel developments in tailored methods and software tools (see, e.g., [66] and [29]) for automatic or semi-automatic extraction of the desired information from the images. Simultaneously, the studies based on the observation of large number of cells also rely on tailored classification methods. These methods are used to pre-select which cells are to be observed by the image analysis techniques at a higher level of detail.

In this study, time-lapse microscopy is used to monitor individual *E. coli* cells over their whole lifetime. The images of cells were collected from the phase contrast to detect the borders of the cells as well as from the confocal microscopy to visualize fluorescently tagged FtsZ proteins. One of the goals of this work was cell segmentation. To do this, we used CellAging [29] and MAMLE [66] tools. During the segmentation process, we also gained the required information about individual cells, e.g., dimensions, location, and orientation. Then, we performed a sample selection technique. To remove unwanted samples, i.e. samples with unknown features, we checked the segmented images one-by-one and then selected those that represent unrealistic values. This was a complex process because many erroneous samples can lie within the standard distribution. However, this task led to an increase in the accuracy of classification.

After that, we tried to find a set of features to distinguish the stages of FtsZ ring formation. Since cells do not have the same shape, size, and orientation in each stage, we had to extract statistical features from each individual sample. In order to perform feature extraction task, cell images were split in three sections, namely, the two poles and midcell at positions 0.25 and 0.75 along the major axis (length normalized to 1). By doing this, from the distribution of fluorescence intensity from FtsZ-GFP along the major cell axis of each cell, two features, mean and standard deviation (std), were measured.

Since the classes of samples were known and also a model can be simply generated by using labelled data, we preferably decided to use supervised learning algorithms. Moreover, we can obtain more specific information by using supervised classification methods in comparison with unsupervised methods (i.e., clustering). Therefore, based on these measured features, 150 samples, 50 samples per class, were labelled by a biologist, as they played the role of a supervisor for our data. Then, we applied a 10-fold cross validation method on the dataset to partition it into training and test sets. We used training set to build the model and then evaluated the accuracy of the model using the test set.

To do classification task, in order to learn the model, we fed training set into three different types of supervised classification methods. We, first, applied Decision Tree (DT) as a simple and non-parametric method, that can cover both binary and multiclass classification problems. The second used method was Support Vector Machine (SVM). SVM is originally known as a binary and maximum margin classifier. In this algorithm, the decision boundary between classes is determined by using labelled samples. Here, to adapt SVM to a multiclass method, we applied One-Against-All (OAA) techniques [103]. The parameters of the model were selected by cross-validation technique (see Section 3.6.4). For the third method, we used Regularized Multinomial Logistic regression (RMLR) algorithm. It generalizes LR to multiclass problems and is a relatively simple and efficient method that applies the maximum likelihood principle for estimating parameters of the model. After building the model, we tested the performance of the model using the test data.

The results of classification methods showed that the regularized MLR outperforms two other methods, DT and SVM, during learning and testing the model. The RMLR obtained the highest average ACC and AUC of 77.34% and 72.13%, respectively. We increased the labelled data to 300 samples, 100 samples per class. Following the increase in the number of labelled samples, in all cases the results improved. However, the RMLR classifier had better performance compared with the other two classifiers, with a 10-fold cross-validated ACC of 79.44% and an AUC score of 75.12%. Based on the obtained results, a conference publication [104] has already been accepted.

In conclusion, we gained reliable results from proposed RMLR supervised method. We found that RMLR is more compatible with the data and measured features, most likely due to being a multi-logit model based on the maximum likelihood principle.

REFERENCES

[1] E. Bi and J. Lutkenhaus, “FtsZ ring structure associated with division in Escherichia coli,” *Nature*, vol. 354, no. 6349, pp. 161–164, 1991.

[2] A. Gupta, J. Lloyd-Price, S. M. D. Oliveira, O. Yli-Harja, A.-B. Muthukrishnan, and A. S. Ribeiro, “Robustness of the division symmetry in *Escherichia coli* and functional consequences of symmetry breaking,” *Phys. Biol.*, vol. 11, no. 6, p. 66005, Nov. 2014.

[3] E. Mulder and C. L. Woldringh, “Actively replicating nucleoids influence positioning of division sites in Escherichia coli filaments forming cells lacking DNA.,” *J. Bacteriol.*, vol. 171, no. 8, pp. 4303–4314, 1989.

[4] T. G. Bernhardt and P. A. J. de Boer, “SlmA, a nucleoid-associated, FtsZ binding protein required for blocking septal ring assembly over chromosomes in E. coli,” *Mol. Cell*, vol. 18, no. 5, pp. 555–564, 2005.

[5] H. Meinhardt and P. a de Boer, “Pattern formation in Escherichia coli: a model for the pole-to-pole oscillations of Min proteins and the localization of the division site.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 98, no. 25, pp. 14202–7, Dec. 2001.

[6] Adams and Errington, “Bacterial cell division: assembly, maintenance and disassembly of the Z ring.,” *Nat. Rev. Microbiol.*, vol. 7, no. 9, pp. 642–53, Sep. 2009.

[7] E. Galli and K. Gerdes, “FtsZ-ZapA-ZapB interactome of Escherichia coli,” *J. Bacteriol.*, vol. 194, no. 2, pp. 292–302, 2012.

[8] X. Ma, D. W. Ehrhardt, and W. Margolin, “Colocalization of cell division proteins FtsZ and FtsA to cytoskeletal structures in living *Escherichia coli* cells by using green fluorescent protein,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 93, no. 23, pp. 12998–3003, 1996.

[9] R. Tsukanov, G. Reshes, G. Carmon, E. Fischer-Friedrich, N. S. Gov, I. Fishov, and M. Feingold, “Timing of Z-ring localization in Escherichia coli,” *Phys. Biol.*, vol. 8, no. 6, p. 66003, Dec. 2011.

[10] S. Rueda, M. Vicente, and J. Mingorance, “Concentration and assembly of the division ring proteins FtsZ, FtsA, and ZipA during the Escherichia coli cell cycle.,” *J. Bacteriol.*, vol. 185, no. 11, pp. 3344–51, Jun. 2003.

[11] H. E. Kubitschek, “Cell volume increase in Escherichia coli after shifts to richer media.,” *J. Bacteriol.*, vol. 172, no. 1, pp. 94–101, Jan. 1990.

[12] İ. E. Nikerel, E. Öner, B. Kirdar, and R. Yildirim, “Optimization of medium composition for biomass production of recombinant Escherichia coli cells using response surface methodology,” *Biochem. Eng.*, vol. 32, no. 1, pp. 1–6, 2006.

[13] A. Sivashanmugam, V. Murray, C. Cui, Y. Zhang, J. Wang, and Q. Li, “Practical protocols for production of very high yields of recombinant proteins using

Escherichia coli," *Protein Sci.*, vol. 18, no. 5, pp. 936–948, 2009.

[14] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*. Garland Science, USA, 2002.

[15] F. R. Blattner, G. I. I. I. Plunkett, A. C. Bloch, T. N. Perna, V. Burland, M. Riley, J. Collado-Vides, D. J. Glasner, K. C. Rode, F. G. Mayhew, J. Gregor, W. N. Davis, A. H. Kirkpatrick, A. M. Goeden, J. D. Rose, B. Mau, and Y. Shao, "The Complete Genome Sequence of Escherichia coli K-12," *Science (80-.).*, vol. 277, no. 5331, pp. 1453–1462, 1997.

[16] Y. Taniguchi, P. J. Choi, G.-W. Li, H. Chen, M. Babu, J. Hearn, A. Emili, and X. S. Xie, "Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells," *Science (80-.).*, vol. 329, no. 5991, pp. 533–538, 2010.

[17] S. Gottesman and M. R. Maurizi, "Regulation by proteolysis: energy-dependent proteases and their targets.," *Microbiol. Mol. Biol. Rev.*, vol. 56, no. 4, pp. 592–621, Dec. 1992.

[18] D. V. Goeddel, D. G. Kleid, F. Bolivar, H. L. Heyneker, D. G. Yansura, R. Crea, T. Hirose, A. Kraszewski, K. Itakura, and A. D. Riggs, "Expression in Escherichia coli of chemically synthesized genes for human insulin.," *Proc. Natl. Acad. Sci.*, vol. 76, no. 1, pp. 106–110, Jan. 1979.

[19] D. Beckett, K. S. Koblan, and G. K. Ackers, "Quantitative study of protein association at picomolar concentrations: The λ phage cI repressor," *Anal. Biochem.*, vol. 196, no. 1, pp. 69–75, 1991.

[20] S. L. Svenningsen, N. Costantino, D. L. Court, and S. Adhya, "On the role of Cro in λ prophage induction," *Proc. Natl. Acad. Sci. United States Am.* , vol. 102, no. 12, pp. 4465–4469, Mar. 2005.

[21] L. Zeng, S. O. Skinner, C. Zong, J. Sippy, M. Feiss, and I. Golding, "Decision Making at a Subcellular Level Determines the Outcome of Bacteriophage Infection," *Cell*, vol. 141, no. 4, pp. 682–691, 2010.

[22] I. Golding and E. C. Cox, "RNA dynamics in live Escherichia coli cells.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 101, no. 31, pp. 11310–11315, 2004.

[23] R. Lutz and H. Bujard, "Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I 2 regulatory elements," *Nucleic Acids Res.*, vol. 25, no. 6, pp. 1203–1210, 1997.

[24] I. Golding, J. Paulsson, S. M. Zawilski, and E. C. Cox, "Real-time kinetics of gene activity in individual bacteria.," *Cell*, vol. 123, no. 6, pp. 1025–1036, 2005.

[25] D. S. Peabody, "The RNA binding site of bacteriophage MS2 coat protein," *EMBO J.*, vol. 12, pp. 595–600, 1993.

[26] M. Kandhavelu, A. Häkkinen, O. Yli-Harja, and A. S. Ribeiro, "Single-molecule dynamics of transcription of the lar promoter," *Phys. Biol.*, vol. 9, p. 26004, 2012.

[27] A. B. Lindner, R. Madden, A. Demarez, E. J. Stewart, and F. Taddei, “Asymmetric segregation of protein aggregates is associated with cellular aging and rejuvenation.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 105, no. 8, pp. 3076–3081, 2008.

[28] A. S. Coquel, J. P. Jacob, M. Primet, A. Demarez, M. Dimiccoli, T. Julou, L. Moisan, A. B. Lindner, and H. Berry, “Localization of Protein Aggregation in Escherichia coli Is Governed by Diffusion and Nucleoid Macromolecular Crowding Effect,” *PLoS Comput. Biol.*, vol. 9, no. 4, 2013.

[29] A. Häkkinen, A.-B. Muthukrishnan, A. Mora, J. M. Fonseca, and A. S. Ribeiro, “CellAging: a tool to study segregation and partitioning in division in cell lineages of Escherichia coli.,” *Bioinformatics*, vol. 29, no. 13, pp. 1708–1709, 2013.

[30] J. Lloyd-Price, A. Gupta, and A. S. Ribeiro, “SGNS2: A Compartmentalized Stochastic Chemical Kinetics Simulator for Dynamic Cell Populations,” *Bioinformatics*, vol. 28, no. 22, pp. 3004–3005, 2012.

[31] C. L. Woldringh, A. Zaritsky, and N. B. Grover, “Nucleoid partitioning and the division plane in Escherichia coli.,” *J. Bacteriol.*, vol. 176, no. 19, pp. 6030–6038, Oct. 1994.

[32] C. L. Woldringh, E. Mulder, P. G. Huls, and N. Vischer, “Toporegulation of bacterial division according to the nucleoid occlusion model,” *Res. Microbiol.*, vol. 142, no. 2, pp. 309–320, 1991.

[33] N. W. Goehring, F. Gueiros-filho, and J. Beckwith, “Premature targeting of a cell division protein to midcell allows dissection of divisome assembly in Escherichia coli Premature targeting of a cell division protein to midcell allows dissection of divisome assembly in Escherichia coli,” pp. 127–137, 2005.

[34] A. G. Marr, R. J. Harvey, and W. C. Trentini, “Growth and division of Escherichia coli.,” *J. Bacteriol.*, vol. 91, no. 6, pp. 2388–2389, Jun. 1966.

[35] F. J. Trueba and C. L. Woldringh, “Changes in cell diameter during the division cycle of Escherichia coli.,” *J. Bacteriol.*, vol. 142, no. 3, pp. 869–878, Jun. 1980.

[36] F. P. Errington, E. O. Powell, and N. Thompson, “Growth Characteristics of Some Gram-negative Bacteria,” *Microbiology*, vol. 39, no. 1, pp. 109–123, 1965.

[37] S. M. Sullivan and J. R. Maddock, “Bacterial division: Finding the dividing line,” *Curr. Biol.*, vol. 10, no. 6, pp. R249–R252, 2000.

[38] J. Cullum and M. Vicente, “Cell growth and length distribution in Escherichia coli.,” *J. Bacteriol.*, vol. 134, no. 1, pp. 330–337, Apr. 1978.

[39] W. D. Donachie, K. J. Begg, and M. Vicente, “Cell length, cell growth and cell division,” *Nature*, vol. 264, pp. 328–333, 1976.

[40] W. D. Donachie and K. J. Begg, “‘Division potential’ in Escherichia coli.,” *J. Bacteriol.*, vol. 178, no. 20, pp. 5971–5976, Oct. 1996.

[41] A. L. Koch, “On Evidence Supporting a Deterministic Process of Bacterial Growth,” *Microbiology*, vol. 43, no. 1, pp. 1–5, 1966.

[42] D. Joseleau-Petit, D. Vinella, and R. D’Ari, “Metabolic Alarms and Cell Division inEscherichia coli,” *J. Bacteriol.*, vol. 181, no. 1, pp. 9–14, 1999.

[43] D. S. Weiss, J. C. Chen, J.-M. Ghigo, D. Boyd, and J. Beckwith, “Localization of FtsI (PBP3) to the Septal Ring Requires Its Membrane Anchor, the Z Ring, FtsA, FtsQ, and FtsL,” *J. Bacteriol.* , vol. 181, no. 2, pp. 508–520, Jan. 1999.

[44] X. Ma and W. Margolin, “Genetic and Functional Analyses of the Conserved C-Terminal Core Domain of Escherichia coli FtsZ,” *J. Bacteriol.*, vol. 181, no. 24, pp. 7531–7544, Dec. 1999.

[45] X.-C. Yu and W. Margolin, “FtsZ ring clusters in min and partition mutants: role of both the Min system and the nucleoid in regulating FtsZ ring localization,” *Mol. Microbiol.*, vol. 32, no. 2, pp. 315–326, Apr. 1999.

[46] R. A. Kerr, H. Levine, T. J. Sejnowski, and W.-J. Rappel, “Division accuracy in a stochastic model of Min oscillations in Escherichia coli,” *Proc. Natl. Acad. Sci. United States Am.* , vol. 103, no. 2, pp. 347–352, Jan. 2006.

[47] M. Thanbichler, “Synchronization of chromosome dynamics and cell division in bacteria.,” *Cold Spring Harb. Perspect. Biol.*, vol. 2, no. 1, p. a000331, Jan. 2010.

[48] J. A. Valkenburg and C. L. Woldringh, “Phase separation between nucleoid and cytoplasm in Escherichia coli as defined by immersive refractometry.,” *J. Bacteriol.*, vol. 160, no. 3, pp. 1151–1157, Dec. 1984.

[49] J. Männik, F. Wu, F. J. H. Hol, P. Bisicchia, D. J. Sherratt, J. E. Keymer, and C. Dekker, “Robustness and accuracy of cell division in Escherichia coli in diverse cell shapes.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 18, pp. 6957–62, May 2012.

[50] S. G. Addinall, C. Cao, and J. Lutkenhaus, “Temperature shift experiments with an ftsZ84(Ts) strain reveal rapid dynamics of FtsZ localization and indicate that the Z ring is required throughout septation and cannot reoccupy division sites once constriction has initiated,” *J. Bacteriol.*, vol. 179, no. 13, pp. 4277–4284, 1997.

[51] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers,” *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[52] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. 2000.

[53] A. R. Webb and K. D. Copsey, *Statistical Pattern*. United Kingdom, 2011.

[54] J. K. Fisher, A. Bourniquel, G. Witz, B. Weiner, M. Prentiss, and N. Kleckner, “Four-dimensional imaging of E. coli nucleoid organization and dynamics in living cells,” *Cell*, vol. 153, no. 4, pp. 882–895, 2013.

[55] R. Heim, D. C. Prasher, and R. Y. Tsien, “Wavelength mutations and posttranslational autoxidation of green fluorescent protein.,” *Proc. Natl. Acad. Sci.*,

vol. 91, no. 26, pp. 12501–12504, 1994.

[56] M. Chalfie, Y. Tu, G. Euskirchen, W. Ward, and D. Prasher, “Green fluorescent protein as a marker for gene expression,” *Science (80-.).*, vol. 263, no. 5148, pp. 802–805, Feb. 1994.

[57] F. Arigoni, K. Pogliano, C. D. Webb, P. Stragier, and R. Losick, “Localization of protein implicated in establishment of cell type to sites of asymmetric division,” *Science (80-.).*, vol. 270, no. 5236, pp. 637–640, 1995.

[58] T. BERNAS, M. ZAREBSKI, R. R. COOK, and J. W. DOBRUCKI, “Minimizing photobleaching during confocal microscopy of fluorescent probes bound to chromatin: role of anoxia and photon flux,” *J. Microsc.*, vol. 215, no. 3, pp. 281–296, Sep. 2004.

[59] D. J. Stephens and V. J. Allan, “Light microscopy techniques for live cell imaging.,” *Science*, vol. 300, no. 5616, pp. 82–6, Apr. 2003.

[60] J. Andersen, S. A. Forst, K. Zhao, M. Inouye, and N. Delihas, “The Function of micF RNA,” *J. Biol. Chem.*, vol. 264, no. 30, pp. 17961–17970, 1989.

[61] T. Misgeld, M. Kerschensteiner, F. M. Bareyre, R. W. Burgess, and J. W. Lichtman, “Imaging axonal transport of mitochondria in vivo,” *Nat. Methods*, vol. 4, no. 7, pp. 559–561, 2007.

[62] M. M. Frigault, J. Lacoste, J. L. Swift, and C. M. Brown, “Live-cell microscopy - tips and tools.,” *J. Cell Sci.*, vol. 122, no. Pt 6, pp. 753–67, Mar. 2009.

[63] V. Magidson and A. Khodjakov, “Circumventing photodamage in live-cell microscopy,” *Methods Cell Biol.*, vol. 114, p. 10.1016/B978-0-12-407761-4.00023-3, 2013.

[64] P. J. Shaw, “Comparison of Widefield/Deconvolution and Confocal Microscopy for Three-Dimensional Imaging,” in *Handbook Of Biological Confocal Microscopy*, J. B. Pawley, Ed. Boston, MA: Springer US, 2006, pp. 453–467.

[65] K. Norman, “Techniques: Intravital microscopy – a method for investigating disseminated intravascular coagulation?,” *Trends Pharmacol. Sci.*, vol. 26, no. 6, pp. 327–332, 2005.

[66] S. Chowdhury, M. Kandhavelu, O. Yli-Harja, and A. S. Ribeiro, “Cell segmentation by multi-resolution analysis and maximum likelihood estimation (MAMLE).,” *BMC Bioinformatics*, vol. 14 Suppl 1, no. 10, p. S8, 2013.

[67] C. C. A. Queimadelas, “Automated segmentation, tracking and evaluation of bacteria in microscopy images,” Faculdade de Ci{ê}ncias e Tecnologia, 2012.

[68] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised leaning,” *Int. J. Comput. Sci.*, vol. 1, no. 2, pp. 111–117, 2006.

[69] M. Grochowski and N. Jankowski, “Comparison of Instance Selection Algorithms II. Results and Comments,” in *Artificial Intelligence and Soft Computing - ICAISC*

2004: 7th International Conference, Zakopane, Poland, June 7-11, 2004. Proceedings, L. Rutkowski, J. H. Siekmann, R. Tadeusiewicz, and L. A. Zadeh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 580–585.

[70] M. Kubat, S. Matwin, and others, "Addressing the curse of imbalanced training sets: one-sided selection," in *ICML*, 1997, vol. 97, pp. 179–186.

[71] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[72] J. W. Grzymala-Busse and M. Hu, "A Comparison of Several Approaches to Missing Attribute Values in Data Mining," in *Rough Sets and Current Trends in Computing: Second International Conference, RSCTC 2000 Banff, Canada, October 16--19, 2000 Revised Papers*, W. Ziarko and Y. Yao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 378–385.

[73] K. Lakshminarayan, S. A. Harp, and T. Samad, "Imputation of Missing Data in Industrial Databases," *Appl. Intell.*, vol. 11, no. 3, pp. 259–275, 1999.

[74] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An Enabling Technique," *Data Min. Knowl. Discov.*, vol. 6, no. 4, pp. 393–423, 2002.

[75] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics Springer, Berlin, 2001.

[76] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[77] S. Safavian and D. Landgrebe, "a Survey of Decision Tree Classifier Methodology1," *IEEE Trans. Syst. Man*, vol. 21, no. 3, pp. 660–674, 1991.

[78] V. N. Vapnik and S. Kotz, *Estimation of dependences based on empirical data*, vol. 40. Springer-Verlag New York, 1982.

[79] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter, "Comparison of view-based object recognition algorithms using realistic 3D models," in *Artificial Neural Networks --- ICANN 96: 1996 International Conference Bochum, Germany, July 16--19, 1996 Proceedings*, C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 251–256.

[80] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[81] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*, 1997, pp. 276–285.

[82] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, 1998, pp. 137–142.

[83] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines,*

Regularization, Optimization, and Beyond. MIT Press, 2002.

[84] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proc. Natl. Acad. Sci.*, vol. 97, no. 1, pp. 262–267, 2000.

[85] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[86] B. Schölkopf and A. J. Smola, *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[87] P. Ruusuvuori, A. Lehmussola, and O. Yli-Harja, "Learning-based method for spot addressing in microarray images," *Proc. SPIE*, vol. 5672. pp. 416–425, 2005.

[88] M. Jordan, J. Kleinberg, and B. Scho, *Pattern Recognition and Machine Learning*. Springer, 2006.

[89] D. W. Hosmer and S. Lemeshow, "Introduction to the logistic regression model," *Appl. Logist. Regression, Second Ed.*, pp. 1–30, 2000.

[90] V. Robles, C. Bielza, P. Larranaga, S. Gonzalez, and L. Ohno-Machado, "Optimizing logistic regression coefficients for discrimination and calibration using estimation of distribution algorithms," *Top*, vol. 16, no. 2, pp. 345–366, 2008.

[91] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent," vol. 33, no. 1, pp. 1–20, 2010.

[92] F. Provost and R. Kohavi, "Guest Editors' Introduction: On Applied Research in Machine Learning," *Mach. Learn.*, vol. 30, no. 2, pp. 127–132, 1998.

[93] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers ROC Graphs : Notes and Practical Considerations for Data Mining Researchers," p. 27, 2003.

[94] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[95] A. Airola, T. Pahikkala, W. Waegeman, B. De Baets, and T. Salakoski, "A comparison of AUC estimators in small-sample studies.," in *MLSB*, 2010, pp. 3–13.

[96] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, 2001.

[97] F. Provost, F. Provost, and P. Domingos, "Well-Trained PETs: Improving

Probability Estimation Trees," 2000.

[98] M. Sahare and H. Gupta, "A review of multi-class classification for imbalanced data," *Int. J. Adv. Comput. ...*, no. 3, pp. 1–5, 2012.

[99] J. Lloyd-Price, S. Startceva, J. G. Chandraseelan, V. Kandavalli, N. Goncalves, A. Häkkinen, and A. S. Ribeiro, "Dissecting the stochastic transcription initiation process in live Escherichia coli," *DNA Res.*, vol. 23, no. 3, pp. 203–214, 2016.

[100] A.-B. Muthukrishnan, A. Martikainen, R. Neeli-Venkata, and A. S. Ribeiro, "In vivo transcription kinetics of a synthetic gene uninvolved in stress-response pathways in stressed Escherichia coli cells," *PLoS One*, vol. 9, no. 9, 2014.

[101] R. Neeli-Venkata, A. Martikainen, A. Gupta, N. Gonçalves, J. Fonseca, and A. S. Ribeiro, "Robustness of the Process of Nucleoid Exclusion of Protein Aggregates in Escherichia coli," *J. Bacteriol.*, vol. 198, no. 6, pp. 898–906, Mar. 2016.

[102] S. M. D. Oliveira, R. Neeli-Venkata, N. S. M. Goncalves, J. A. Santinha, L. Martins, H. Tran, J. Mäkelä, A. Gupta, M. Barandas, A. Häkkinen, J. Lloyd-Price, J. M. Fonseca, and A. S. Ribeiro, "Increased cytoplasm viscosity hampers aggregate polar segregation in Escherichia coli," *Mol. Microbiol.*, vol. 99, no. 4, pp. 686–699, Feb. 2016.

[103] G. Anthony, H. Gregg, and M. Tshilidzi, "Image classification using SVMs: one-against-one vs one-against-all," *arXiv Prepr. arXiv0711.2914*, 2007.

[104] M. Zare, R. Neeli-Venkata, L. Martins, S. Peltonen, U. Ruotsalainen, and A. S. Ribeiro, "Automatic Classification of Z-Ring Formation Stages at the Single Cell," 2017.